

## Tabla de contenido

|   |    |
|---|----|
| Programación .....  | 4  |
| Conceptos .....   | 4  |
| Actividades .....   | 4  |
| Contexto .....  | 4  |
| Cambiar de una actividad a otra .....                                   | 5  |
| Funciones .....   | 5  |
| OnCreate .....  | 5  |
| OnResume .....  | 5  |
| OnPause .....   | 6  |
| Variables y archivos .....  | 6  |
| Id .....  | 6  |
| Modo portrait y landscape .....   | 7  |
| Usar un XML en otra actividad .....                                     | 7  |
| Guardar datos de forma temporal (persistente) – SharedPreferences ..... | 7  |
| Leer datos (persistentes) – SharedPreferences .....                     | 8  |
| Transferir datos entre actividades .....                                | 8  |
| Recibir datos desde otra actividad .....                                | 8  |
| Android Manifest .....  | 9  |
| XML .....   | 10 |
| Establecer un fondo en el XML .....                                     | 10 |
| Espacios .....  | 10 |
| Objetos y procesos .....  | 10 |
| TextView .....  | 10 |
| EditText .....  | 10 |
| ScrollView .....  | 10 |
| Tablas .....  | 10 |
| AlertDialog .....   | 12 |
| Floating Action Button .....  | 13 |
| Barra de herramientas .....   | 13 |
| ToolBar .....   | 13 |
| Menú .....  | 15 |

|   |    |
|---|----|
| Imágenes .....  | 16 |
| Radio Group .....   | 17 |
| Ocultar botones y objetos de la vista del usuario .....           | 17 |
| Fragmentos .....  | 18 |
| Hilos.....  | 18 |
| Sonidos .....   | 18 |
| MediaPlayer .....   | 18 |
| Funciones .....   | 19 |
| Salir de la aplicación .....                                      | 19 |
| Mostrar mensajes simples con 'Toast' .....                        | 19 |
| OnBackPressed – Botón hacia atrás.....                            | 19 |
| RequestOrientation – Cambiar orientación .....                    | 19 |
| Move Task To Back – no regresa en la actividad.....               | 20 |
| getString .....   | 20 |
| TextUtils .....   | 20 |
| Log para mensajes en consola.....                                 | 20 |
| Eventos .....   | 20 |
| Pulsación larga - OnLongClickListener.....                        | 20 |
| Pulsación corta – OnClickListener .....                           | 20 |
| On Activity Result – Si esperamos que una actividad termine ..... | 21 |
| OnTouchEvent .....  | 21 |
| Hardware.....   | 22 |
| Cámara .....  | 22 |
| Acelerómetro .....  | 23 |
| Servicios.....  | 24 |
| Transacciones de red.....   | 24 |
| Lectura y escritura de archivos .....                             | 24 |
| Música .....  | 24 |
| Fragmentos .....  | 26 |
| Manejador de Fragmentos.....                                      | 28 |
| Framento de menú.....   | 29 |
| Fragmento de música .....   | 31 |
| Fragmento de acelerómetro (nivelador).....                        | 35 |

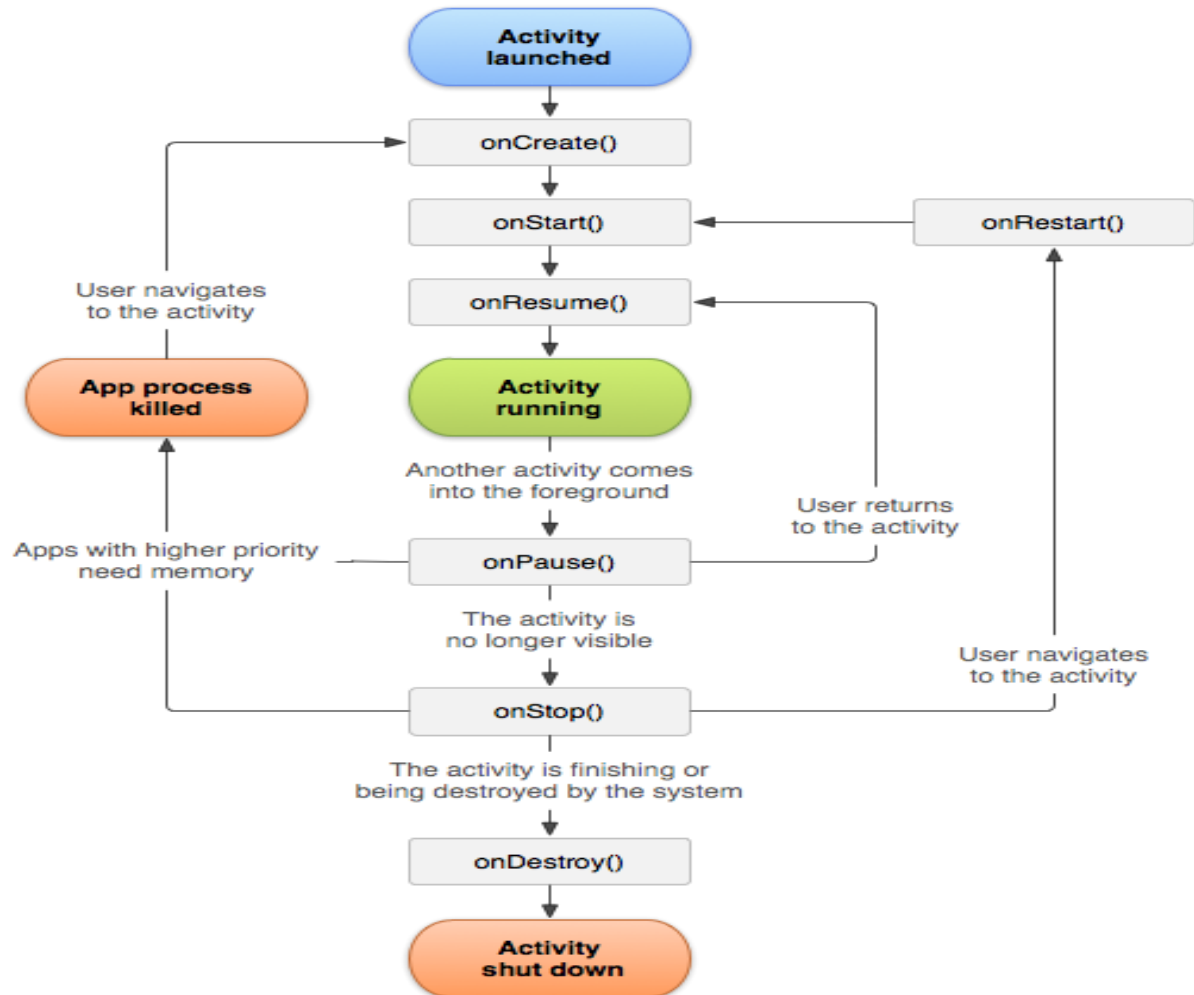
|   |    |
|---|----|
| Fragmento de linterna.....                                      | 42 |
| Firestore .....   | 45 |
| signOut .....   | 45 |
| Email verification.....   | 46 |
| Registrar .....   | 47 |
| Bases de datos.....   | 56 |
| Create .....  | 56 |
| Read.....   | 56 |
| Update.....   | 56 |
| Delete .....  | 56 |
| Juegos.....   | 56 |
| Set content view – establecer una vista desde un programa ..... | 56 |
| Dibujo del juego .....  | 56 |
| Carpeta RES .....   | 59 |
| Drawable .....  | 59 |
| Crear Objetos reutilizables.....                                | 59 |
| Layout.....   | 61 |
| MipMap.....   | 63 |
| Values .....  | 63 |
| colors.xml .....  | 63 |
| strings.xml .....   | 64 |
| styles.xml.....   | 64 |
| dimens.xml .....  | 66 |
| Raw .....   | 67 |
| APIs.....   | 67 |
| Google Maps .....   | 67 |
| Clases.....   | 67 |
| Tablas .....  | 67 |
| Fecha .....   | 75 |

# Programación

## Conceptos

### Actividades

Las actividades son las pantallas que se mostrarán en el teléfono, internamente siguen el siguiente ciclo de vida:



Heredan de Activity, se puede usar AppCompatActivity  
Como sustituto

### Contexto

Indica el origen (La clase Actividad). Y toma el objeto origen como referencia

## Cambiar de una actividad a otra

Al presionar la view se ejecuta su evento asociado

```
public void verImagen(View view){
    Intent intent = new Intent(this,ExtraActivity.class);
    startActivity(intent);
}
```

## Funciones

Las funciones pueden ser personales o sobreescritas. Si la función responde a un evento entonces probablemente deba enviarse el objeto como parámetro. Ejemplo para una función del botón salir que responde cuando es pulsado:

```
public void salir(View view){
    finish();
}
```

## OnCreate

**El layout es la parte del diseño en XML**

**SavedInstances es usado para utilizar persistencia de la información**

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_horario);
}
```

## OnResume

Pone a la escucha un sensor

```
public void onResume() {

    super.onResume();

    manager.registerListener(this,sensor,SensorManager.SENSOR_DELAY_GAME);
}

public void onResume() {
    super.onResume();
    leeRecord();
}
```

```

public void leeRecord() {
    SharedPreferences datos =
    PreferenceManager.getDefaultSharedPreferences(this);
    record = datos.getInt("record", 0);

    TextView result = (TextView) findViewById(R.id.score);
    result.setText("Record: " + record);
}

```

## OnPause

Le quita la escucha al sensor

```

public void onPause() {
    super.onPause();
    manager.unregisterListener(this);
}

```

## Variables y archivos

El nombramiento de variables sigue un formato libre.

El nombramiento de objetos mantiene el nombre original del objeto al inicio. Ejemplo para nombrar un objeto de la clase EditText que corresponde a un título:

```

editTextTituloPrincipal = findViewById(R.id.editTextTituloPrincipal);

```

Las clases empiezan con mayúscula

Las actividades siguen los principios de las clases pero con 'Activity' al final. Ejemplo:

AgregarActivity.java

Los layouts referentes a cada actividad se nombran en minúsculas "al igual" que el Activity principal. Ejemplo: activity\_agregar.xml

## Id

Es el identificador de un objeto

Del parámetro podemos obtener su id

```

(View view){
    if(view.getId()

```

Es recomendable usar arreglos que almacenen los IDs

```

ImageView imagen = (ImageView) findViewById(casillas[fila][columna]);

if(partida.jugador == 1) {

```

```

        imagen.setImageResource(R.drawable.circulo);
    }
    else{
        imagen.setImageResource(R.drawable.aspa);
    }
}

```

## Modo portrait y landscape

Esto fuerza a que esté en landscape

```

<activity android:name=".MainActivity" android:theme="@style/Theme.AppCompat.NoActionBar"
android:screenOrientation="landscape">

    <intent-filter>

        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />

    </intent-filter>

</activity>

```

## Usar un XML en otra actividad

Para usar un xml desde otra actividad debemos 'extender' de Activity y con la funcion onCreate establecemos el xml

```

public class ViewInfo extends Activity {

    public void onCreate(Bundle savedInstanceState){

        super.onCreate(savedInstanceState);

        setContentView(R.layout.informacion);

    }

}

```

## Guardar datos de forma temporal (persistente) – SharedPreferences

```

public void guardarRecord() {
    SharedPreferences datos =
    PreferenceManager.getDefaultSharedPreferences(this);
    SharedPreferences.Editor editor = datos.edit();
    editor.putInt("record", record);
    editor.apply();
}

```

### Leer datos (persistentes) – SharedPreferences

```
public void leeRecord() {
    SharedPreferences datos =
    PreferenceManager.getDefaultSharedPreferences(this);
    record = datos.getInt("record", 0);

    TextView result = (TextView) findViewById(R.id.score);
    result.setText("Record: " + record);
}
```

### Transferir datos entre actividades

Para transferir datos, agregamos al intent la información

```
public void test(View view){

    Intent intent_test = new Intent(this, ViewTesteo.class);

    if(!num1.getText().toString().equals("") && !num2.getText().toString().equals("")) {

        intent_test.putExtra("num1", Integer.parseInt(num1.getText().toString()));

        intent_test.putExtra("num2", Integer.parseInt(num2.getText().toString()));

        num1.setText("");

        num2.setText("");

        startActivity(intent_test);

    }

}
```

### Recibir datos desde otra actividad

Si transferimos datos, podemos recolectarlos con la clase Bundle

```
public void onCreate(Bundle savedInstanceState){

    super.onCreate(savedInstanceState);

    setContentView(R.layout.testeo);

    Bundle datos = getIntent().getExtras();

    int num1 = datos.getInt("num1"), num2 = datos.getInt("num2");
```



```

TextView suma = (TextView)findViewById(R.id.textView_testeo);

suma.setText("La suma es: " + (num1+num2));

}

```

## Android Manifest

Podemos también editar la forma en que se mostrará

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.game.magody.fragments">

    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.FLASHLIGHT"/>

    <uses-feature android:name="android.hardware.camera"/>
    <uses-feature android:name="android.hardware.camera.flash"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:screenOrientation="portrait"
            android:theme="@style/Theme.AppCompat">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
            />
            </intent-filter>
        </activity>
        <activity android:name=".HerramientasActivity"
            android:screenOrientation="portrait"/>

        <service android:name=".ServicioMusica" android:enabled="true"/>
    </application>

</manifest>

    <activity android:name=".ExtraActivity"></activity>
</application>

```

Al crear una actividad se la debe agregar al AndroidManifest

```

<activity android:name=".ViewInfo"

    android:theme="@style/Theme.AppCompat.Dialog.Alert" />

```

## XML

```
<LinearLayout 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"
    android:padding="@dimen/padding_pantalla"
    android:weightSum = "1"
    android:orientation="horizontal"
    android:background="@drawable/fondo"

    tools:context="com.game.magody.tictactoe.MainActivity">
```

Establecer un fondo en el XML

```
android:background="@drawable/fondo"
```

Espacios

```
<Space
    style="@style/espacio"/>
```

## Objetos y procesos

TextView

Usado como label para mostrar titulos, etc

Si no se lo va a usar totalmente se puede establecer de la siguiente manera

```
((TextView) findViewById(R.id.tv_info)).setText("Dias totales de la
actividad: " + dias_iniciales_disponibles);
```

EditText - Son espacios que puede modificar el usuario

ScrollView

Permite agregar textos muy largos y realizar scroll para leerlo. Un subitem textView debe ser agregado

Tablas

```
private void iniciarTabla() {

    metas_objetivos = new ArrayList<>();
    metas_presicion = new ArrayList<>();
    tableLayout = findViewById(R.id.table);
    String[] header = {"Titulo", "Inicio", "Fin"};
    tabla_metas = new
    TablaMetas(tableLayout, getApplicationContext(), this);
```

```

        tabla_metas.crearHeader(header);

        tabla_metas.agregarFilasData(new ArrayList<String[]>() {new
String["a11","a12"], new String["a21","a22"]}
);

        tabla_metas.backgroundHeader(0xffeaf6fc);
        tabla_metas.backgroundData(0xff62ace5,0xff9fcef2);
        tabla_metas.lineColor(Color.BLACK);
        tabla_metas.textColorData(Color.BLACK);
        tabla_metas.textColorHeader(Color.BLACK);
    }

    public void crearHeader(String[] header) {
        this.header = header;
        TableRow tabla_fila = new TableRow(context);
        for (String h : header) { //total de datos dentro del arreglo
            TextView tv_celda = new TextView(context);
            tv_celda.setGravity(Gravity.CENTER);
            tv_celda.setTextSize(25);
            tv_celda.setText(h);
            tabla_fila.addView(tv_celda, parametrosFila());
        }
        tableLayout.addView(tabla_fila);
    }

    public void agregarFilasData(ArrayList<String[]> data) {
        this.datos = data;
        String info;
        for (int indexR = 1; indexR <= data.size(); indexR++) {
            //desde la 1 porque en la cero esta el encabezado
            TableRow tabla_fila = new TableRow(context);
            tabla_fila.setOnLongClickListener(new View.OnLongClickListener()
            {
                @Override
                public boolean onLongClick(View v) {
                    meta = ((TextView) ((TableRow)
v).getChildAt(0)).getText().toString();
                    actividad_main.alerta_eliminar.show();
                    return true;
                }
            });

            tabla_fila.setOnClickListener(this);
            for (int indexC = 0; indexC < header.length; indexC++) {
                TextView tv_celda = new TextView(context);
                tv_celda.setGravity(Gravity.CENTER);
                tv_celda.setTextSize(25);
                String[] row = data.get(indexR - 1); // en la posicion cero
estará la informacion
                info = (indexC < row.length) ? row[indexC] : ""; // si solo
tengo tres columnas en las demas se agrega un vacio
                tv_celda.setText(info);
                tabla_fila.addView(tv_celda, parametrosFila());
            }
        }
    }

```

```

        tableLayout.addView(tabla_fila);
    }
}

```

## AlertDialog

Sirve para advertir y realizar acciones según la opción del usuario

```

public AlertDialog alerta_eliminar;

private void inicializarAlertaEliminarMeta() {

    final AlertDialog.Builder builder = new
    AlertDialog.Builder(this);
    builder.setTitle(R.string.dialogo_titulo_main);
    builder.setMessage(R.string.dialogo_main);

    builder.setPositiveButton(R.string.dialogo_si, new
    DialogInterface.OnClickListener() {

        public void onClick(DialogInterface dialog, int which) {

            String archivo_filtro =
            Archivo.leerFiltroMeta(MainActivity.archivo_main,builder.getContext
            (),TableDinamic.meta);

            Archivo.escribirReArchivo(MainActivity.archivo_main,archivo_filtro
            ,builder.getContext());

            File file_pasos =
            getBaseContext().getFileStreamPath(TableDinamic.meta+".txt");
            if(file_pasos.exists())
                file_pasos.delete();

            tableLayout.removeAllViews();
            iniciarTabla();

            dialog.dismiss();
        }
    });

    builder.setNegativeButton(R.string.dialogo_no, new
    DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {

            // Do nothing
            dialog.dismiss();
        }
    });
}

```

```

        alerta_eliminar = builder.create();
    }

```

Se las invoca también desde el evento principal de otro objeto:

```

public void reset(View view) {
    alerta_reset.show();
}

```

## Floating Action Button

Suele venir por defecto creado en algunos templates

```

final FloatingActionButton fab = findViewById(R.id.fab_agregar);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new
Intent(fab.getContext(), AgregarActivity.class);
        startActivityForResult(intent, 1);
    }
});

```

## Barra de herramientas

Proceso para agregar una toolbar

Creamos la variable toolbar

```
private Toolbar bar;
```

le otorgamos la vista previamente creada

```
bar = (Toolbar)findViewById(R.id.tool_bar);
```

```
    setSupportActionBar(bar);
```

y la agregamos, todo esto en onCreate

## ToolBar

```

private Toolbar bar;
bar = (Toolbar)findViewById(R.id.tool_bar);
setSupportActionBar(bar);

```

```

Toolbar bar = findViewById(R.id.tool_bar);
setSupportActionBar(bar);
if(getSupportActionBar() != null){
    getSupportActionBar().setTitle(R.string.registroTituloBar);
}

```

```

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

```

Asociado a un menu.xml de layouts creado anteriormente

```

@Override
public boolean onCreateOptionsMenu(Menu menu_main) {
    getMenuInflater().inflate(R.menu.menu_main, menu_main);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem opcion_menu) {
    int id = opcion_menu.getItemId();

    if(id == R.id.btnconfiguracion) {
        mostrarViewConfiguracion(null);
        return true;
    }
    else if(id == R.id.btninfo) {
        mostrarViewInfo(null);
        return true;
    }

    return super.onOptionsItemSelected(opcion_menu);
}

```

menu.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".MainActivity">
    <item android:id="@+id/btnconfiguracion"
        android:title="Configuracion"
        android:icon="@android:drawable/ic_menu_preferences"
        app:showAsAction="never" />
    <item android:id="@+id/btninfo"
        android:title="Informacion"
        android:icon="@android:drawable/ic_menu_info_details"
        android:orderInCategory="10"
        app:showAsAction="ifRoom"
        />
    <item android:id="@+id/menu_buscar"
        android:title="Buscar"
        android:icon="@android:drawable/ic_menu_search"
        app:showAsAction="always"
        android:orderInCategory="55"
        />
</menu>

```

```
getActionBar().hide();  
getSupportActionBar().hide();
```

Quitar las toolbar

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">  
    <!-- Customize your theme here. -->  
    <item name="colorPrimary">@color/colorPrimary</item>  
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
    <item name="colorAccent">@color/colorAccent</item>  
</style>
```

## Menú

Para ubicar un menú se crea un res especial de menu y con la siguiente mecánica se lo implementa

@Override

```
public boolean onCreateOptionsMenu(Menu menu_main){  
    getMenuInflater().inflate(R.menu.menu_main,menu_main);  
    return true;  
}
```

@Override

```
public boolean onOptionsItemSelected(MenuItem opcion_menu){  
    int id = opcion_menu.getItemId();  
  
    if(id == R.id.btnconfiguracion){  
        mostrarViewConfiguracion(null);  
        return true;  
    }  
  
    else if(id == R.id.btninfo){
```

```

        mostrarViewInfo(null);

        return true;
    }

    return super.onOptionsItemSelected(opcion_menu);

}

```

```

android:icon="@android:drawable/ic_menu_mapmode" //iconos
app:showAsAction="never" //no lo muestra como icono en la barra
app:showAsAction="ifRoom" //lo muestra solo si hay espacio
app:showAsAction="always" //siempre lo muestra como icono en la barra

```

material design dice cambiar las actionBar por toolbar

si usamos toolbar, hay que cambiar el tema en style para quitar actionBar

## Imágenes

Mecánica para agregar una imagen

```

imagen = (ImageView) findViewById(casillas[i][j]);

imagen.setImageResource(R.drawable.casilla);

```

En este caso las id están almacenadas en un vector de enteros

En el xml

```

<ImageView
    android:id="@+id/a31"
    style="@style/casilla"
    android:src="@drawable/circulo"
/>

```



## Radio Group

Manejo de un radio group para elegir una sola opción

```
RadioGroup dificultades = (RadioGroup) findViewById(R.id.dificultades);
```

```
int id_dificultad = dificultades.getCheckedRadioButtonId();
```

```
int dificultad = 0;
```

```
if(id_dificultad == R.id.dificultad_normal){  
    dificultad = 1;  
}  
  
else if(id_dificultad == R.id.dificultad_imposible){  
    dificultad = 2;  
}
```

XML:

```
<RadioGroup  
    android:id="@+id/dificultades"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="0.5"  
    android:gravity="center_vertical|center_horizontal">  
  
    <RadioButton  
        android:id="@+id/dificultad_facil"  
        style="@style/radio"  
        android:text="@string/dificultad_facil" />  
  
    <RadioButton  
        android:id="@+id/dificultad_normal"  
        style="@style/radio"  
        android:checked="true" <!--pone por defecto-->  
        android:text="@string/dificultad_normal" />  
  
    <RadioButton  
        android:id="@+id/dificultad_imposible"  
        style="@style/radio"  
        android:text="@string/dificultad_imposible" />  
</RadioGroup>
```

Ocultar botones y objetos de la vista del usuario

Ocultar botones que ya no se usan

//busca los botones especificados y los oculta

```
((Button) findViewById(R.id.unjugador)).setEnabled(false);  
((RadioGroup) findViewById(R.id.dificultades)).setAlpha(0);  
((Button) findViewById(R.id.dosjugadores)).setEnabled(false);
```

## Fragmentos

Permite la reutilización de varios diseños, por ejemplo: menú

## Hilos

```
private Handler temporizador;  
temporizador = new Handler();  
temporizador.postDelayed(hilo, 2000);  
private Runnable hilo = new Runnable() {  
  
    @Override  
    public void run() {  
  
        if(!partida.movimientoBola()){  
            //no si se acaba el juego  
            partida.invalidate(); // elimina el contenido de ImageView y  
            llama de nuevo a onDraw  
            //pausa de borrar y pintar  
            temporizador.postDelayed(hilo, 1000/fps);  
        }  
        else{  
            fin();  
        }  
    }  
};
```

Se debe establecer esto cuando se termine una actividad

```
temporizador.removeCallbacks(hilo); // limpia el buffer que pudo quedar
```

## Sonidos

### MediaPlayer

```
MediaPlayer golpe, game_over;
```

```
golpe = MediaPlayer.create(this, R.raw.hit_ball);  
game_over = MediaPlayer.create(this, R.raw.game_over);
```

Reproducir sonidos.

```
golpe.start();
```

```
game_over.start();
```

## Funciones

### Salir de la aplicación

```
public void salir(View view){  
  
    finish();  
  
}
```

### Mostrar mensajes simples con 'Toast'

Mecanica para imprimir mensajes con Toast

```
public void imprimir(String msg,int delay){  
  
    Toast toast = Toast.makeText(this,msg,delay);  
  
    toast.setGravity(Gravity.CENTER,0,0);  
  
    toast.show();  
  
}
```

### OnBackPressed – Botón hacia atrás

```
public void onBackPressed() {  
    finish(); // código personalizado  
}
```

También se la puede invocar desde otro botón extra

```
public void volver(View view) {  
    onBackPressed();  
}
```

### RequestOrientation – Cambiar orientación

```
getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
```

Move Task To Back – no regresa en la actividad

```
public void onBackPressed() {  
    // asi no regresa en la actividad  
    moveTaskToBack(true);  
}
```

getString

getString(R.string.mensajeEmailVacio)

TextUtils

TextUtils.isEmpty(password)

Log para mensajes en consola

Imprimir en consola

El tag se define como final al inicio de toda la clase y se usa para saber de donde vino el error

Log.d(TAG, "createUserWithEmail:success");

Eventos

Pulsación larga - OnLongClickListener

```
TableRow tabla_fila = new TableRow(context);  
tabla_fila.setOnLongClickListener(new View.OnLongClickListener() {  
    @Override  
    public boolean onLongClick(View v) {  
        meta = ((TextView) ((TableRow)  
v).getChildAt(0)).getText().toString();  
        actividad_main.alerta_eliminar.show();  
        return true;  
    }  
});
```

Pulsación corta – OnClickListener

```
tabla_fila.setOnClickListener(this);  
for(int indexC = 0; indexC < header.length; indexC++){  
    TextView tv_celda = new TextView(context);  
    tv_celda.setGravity(Gravity.CENTER);  
    tv_celda.setTextSize(25);  
    info = (indexC < item.length) ? item[indexC] : ""; // si solo tengo tres  
columnas en las demas se agrega un vacio  
    tv_celda.setText(info);  
    tabla_fila.addView(tv_celda, parametrosFila());  
}
```

On Activity Result – Si esperamos que una actividad termine

Si deseamos usar información de la otra actividad, al cambiarnos a ella usamos `startActivityResult()`

En este caso agregado a un botón flotante

```
final FloatingActionButton fab = findViewById(R.id.fab_agregar);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new
Intent(fab.getContext(),AgregarActivity.class);
        startActivity(intent,1);
    }
});
```

En la misma aplicación principal podemos gestionar según el número de petición. El 1 indica que es la petición número 1 como se estableció antes.

```
protected void onActivityResult(int peticion, int codigo, Intent intent){

    if(codigo == RESULT_OK){
        if(peticion == 1){
            int len = intent.getIntExtra("len",-1);
            for(int i=0;i<len;i++){
                String[] it = intent.getStringArrayExtra("item"+i);
                tabla_metas.agregarFilaInfo(it);
            }
        }
    }
}
```

Desde la otra actividad cuando va a terminar

```
public void onBackPressed(){

    if(!items.isEmpty()){
        Intent intent = new Intent();
        int len = items.size();
        intent.putExtra("len",len);
        for(int i=0;i<len;i++)
            intent.putExtra("item"+i,items.get(i));

        setResult(RESULT_OK,intent);
    }
    else
        setResult(RESULT_CANCELED);

    super.onBackPressed();
}
```

OnTouchEvent

Obtener posición presionada:

```
int x = (int)event.getX(), y = (int)event.getY();
```

```
public boolean onTouchEvent(MotionEvent event){  
    //evento que reacciona cuando se topa la pantalla  
  
    int x = (int)event.getX(), y = (int)event.getY();  
  
    if(partida.toque(x, y))  
    {  
        botes++;  
        golpe.start(); //suena el mediaplayer de golpe  
    }  
  
    return false;  
}
```

## Hardware

### Cámara

```
public interface ManejaFlashCamara {  
  
    void enciendeApaga(boolean estado_camara);  
  
}  
  
public class HerramientasActivity extends AppCompatActivity implements  
ComunicaMenu, ManejaFlashCamara {  
  
    private CameraManager camara;  
    private String id_camara;
```

Dentro de onCreate de la clase invocadora

```
camara = (CameraManager) getSystemService(Context.CAMERA_SERVICE);  
  
try{  
    id_camara = camara.getCameraIdList()[0];  
}catch (Exception e){  
    e.printStackTrace();  
}
```

Y se sobrescribe

```
@Override
public void enciendeApaga(boolean estado_camara) {
    Toast toast;
    String salida = "Flash ";

    try{
        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) // marshmallow
            6.9 mayor a esa
                camara.setTorchMode(id_camara,!estado_camara);

    }catch (Exception e){
        e.printStackTrace();
    }

    salida += (estado_camara)? "apagado": "activado";

    toast = Toast.makeText(this,salida, Toast.LENGTH_SHORT);
    toast.show();
}
```

Permisos

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

<uses-feature android:name="android.hardware.camera"/>
<uses-feature android:name="android.hardware.camera.flash"/>
```

Acelerómetro

```
public class NivelFragment extends Fragment implements
SensorEventListener

private SensorManager manager;
private Sensor sensor;

private NivelPantalla pantalla;

public NivelFragment() {
    // Required empty public constructor
}

public final void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    manager = (SensorManager)
getActivity().getSystemService(Context.SENSOR_SERVICE);
```

```

        sensor = manager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);

        int lado = getResources().getDimensionPixelSize(R.dimen.maximo);
        pantalla = new NivelPantalla(getActivity(),lado);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {

        pantalla.angulos(event.values);
    }


    public void onResume() {

        super.onResume();

        manager.registerListener(this,sensor,SensorManager.SENSOR_DELAY_GAME);
    }

    public void onPause() {
        super.onPause();
        manager.unregisterListener(this);
    }

```

## Servicios

Realiza tareas en Segundo plano

Transacciones de red

Lectura y escritura de archivos

Música

Eventos de encendido y apagado

```

private void enciendeMusica(){

        boton_musica.setImageResource(R.drawable.musica2);
        Intent intent = new Intent(getActivity(), ServicioMusica.class);
        getActivity().startService(intent);
    }

    private void apagaMusica(){

```



```

        boton_musica.setImageResource(R.drawable.musica);
        Intent intent = new Intent(getActivity(),ServicioMusica.class);
        getActivity().stopService(intent);

    }

```

ServicioMusica.java:

```

package com.game.magody.fragments;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.IBinder;
import android.support.annotation.Nullable;

/**
 * Created by User on 25/03/2018.
 */

public class ServicioMusica extends Service {

    MediaPlayer reproduutor;

    public void onCreate() {
        super.onCreate();
        reproduutor = MediaPlayer.create(this,R.raw.jack);
        reproduutor.setLooping(false);
        reproduutor.setVolume(100,100);

    }

    public int onStartCommand(Intent intent, int flags, int startId){

        reproduutor.start();

        return START_STICKY;

    }

    public void onDestroy() {
        super.onDestroy();
        if(reproduutor.isPlaying()){
            reproduutor.stop();
        }
        reproduutor.release();
        reproduutor = null;

    }
}

```

```

    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}

```

## Fragmentos

Funciona como una actividad, y genera también un layout con su diseño

Es necesario especificar si debe adaptarse al contenido o ajustarse al del padre

```

android:layout_width="match_parent"
android:layout_height="wrap_content"

```

Su onCreate carga el fragment en la actividad que lo aloja, devuelve su view

```

@Override
//buscar inflater en la api de android
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_menu, container, false);
}

```

Se lo puede almacenar en un objeto View, luego llamar a la interfaz y decirle el botón pulsado.

```

public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View menu = inflater.inflate(R.layout.fragment_menu, container,
false);

    if(getArguments() != null)
        boton_pulsado = getArguments().getInt("boton_pulsado");
    else
        boton_pulsado = -1;
}

```

//le agrega eventos a cada botón del menu

```

ImageButton boton_menu;
for(int i=0; i<BOTONES_MENU.length;i++){

    boton_menu = (ImageButton) menu.findViewById(BOTONES_MENU[i]);

    if(boton_pulsado == i){

boton_menu.setImageResource(BOTONES_ILUMINADOS[boton_pulsado]);
    }
}

```

```

        final int boton_pulsado = i;
        boton_menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Activity esta_actividad = getActivity();

                // casting a la interface, le indica en que clase
                estamos, Llamo a la función de la actividad en la que se encuentra!!!
                ((ComunicaMenu)esta_actividad).menu(boton_pulsado);
            }
        });

    }

    return menu;
}

```

Incluir un fragmento en el ActivityMain

Es necesario una interfaz con las funciones que se puedan usar en las actividades:

```

public interface ComunicaMenu {

    public void menu(int boton_pulsado);

}

public class MainActivity extends AppCompatActivity implements
ComunicaMenu

```

Y sobreescribimos la función:

```

@Override
public void menu(int boton_pulsado) {

    Intent intent = new Intent(this, HerramientasActivity.class);
    intent.putExtra("boton_pulsado", boton_pulsado);
    startActivity(intent);

}

```

XML:

```

<fragment
    android:id="@+id/menu"

```

```

        android:name="NOMBRE DEL FRAGMENTO" // los nombres deberían incluirse
con autocompletado
        tools:layout="@layout/fragment_menu"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

```

## Manejador de Fragmentos

En la siguiente actividad que tiene el fragment rescatamos el botón pulsado.

Además es posible guardar los fragmentos en un arreglo

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_herramientas);

    fragments = new Fragment[3];
    fragments[0] = new LinternaFragment();
    fragments[1] = new NivelFragment();
    fragments[2] = new MusicaFragment();

    Bundle extras = getIntent().getExtras();
    menu(extras.getInt("boton_pulsado"));

    camara = (CameraManager) getSystemService(Context.CAMERA_SERVICE);

    try{

        id_camara = camara.getCameraIdList()[0];
    } catch (Exception e){
        e.printStackTrace();
    }

}

```

## Transacción y reemplazo de fragmentos

```

@Override
public void menu(int boton_pulsado) {

    FragmentManager manejador = getSupportFragmentManager();
    FragmentTransaction transaccion = manejador.beginTransaction();

    Fragment iluminado = new MenuFragment();
    Bundle datos = new Bundle();
    datos.putInt("boton_pulsado", boton_pulsado);
}

```

```

        iluminado.setArguments(datos); //le cambia el color a una imagen

        transaccion.replace(R.id.menu, iluminado); //cambia de fragmento
        transaccion.replace(R.id.contenedor, fragments[boton_pulsado]);
        cambia de fragmento

        transaccion.commit();
    }

```

El contenedor es manejado en el xml

```

<RelativeLayout
    android:id="@+id/contenedor"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

</RelativeLayout>

```

Fragmento de menú

JAVA:

```

package com.game.magody.fragments;

import android.app.Activity;
import android.media.Image;
import android.os.Bundle;
import android.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;

/**
 * A simple {@link Fragment} subclass.
 */
public class MenuFragment extends Fragment {

    private final int[] BOTONES_MENU = {R.id.linterna,R.id.nivel,
    R.id.musica };
    private final int[] BOTONES_ILUMINADOS = {R.drawable.linterna2,
    R.drawable.nivel2, R.drawable.musica2};
    private int boton_pulsado;

```

```

public MenuFragment() {
    // Required empty public constructor
}

@Override
//buscar inflater en la api de android
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                        Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View menu = inflater.inflate(R.layout.fragment_menu, container,
false);

    if(getArguments() != null)
        boton_pulsado = getArguments().getInt("boton_pulsado");
    else
        boton_pulsado = -1;

    ImageButton boton_menu;
    for(int i=0; i<BOTONES_MENU.length;i++){

        boton_menu = (ImageButton)
menu.findViewById(BOTONES_MENU[i]);

        if(boton_pulsado == i){
            boton_menu.setImageResource(BOTONES_ILUMINADOS[boton_pulsado]);
        }

        final int boton_pulsado = i;
        boton_menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Activity esta_actividad = getActivity();

                // casting a la interface
                ((ComunicaMenu)esta_actividad).menu(boton_pulsado);
            }
        });

    }

    return menu;
}
}

```

XML:

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_gravity="center_horizontal"
    tools:context="com.game.magody.fragments.MenuFragment">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:gravity="center_horizontal"
        android:orientation="horizontal"
        android:weightSum="1">

        <ImageButton
            android:id="@+id/linterna"
            style="@style/botones"
            android:src="@drawable/linterna" />

        <Space
            style="@style/espacio"/>

        <ImageButton
            android:id="@+id/nivel"
            style="@style/botones"
            android:src="@drawable/nivel" />

        <Space
            style="@style/espacio"/>

        <ImageButton
            android:id="@+id/musica"
            style="@style/botones"
            android:src="@drawable/musica" />

    </LinearLayout>

</FrameLayout>

```

MAIN XML:

```

<fragment
    android:id="@+id/menu"
    android:name="com.game.magody.fragments.MenuFragment"
    tools:layout="@layout/fragment_menu"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

```

Fragmento de música

ServicioMusica.java:

```
package com.game.magody.fragments;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.IBinder;
import android.support.annotation.Nullable;

/**
 * Created by User on 25/03/2018.
 */

public class ServicioMusica extends Service {

    MediaPlayer reproductor;

    public void onCreate() {
        super.onCreate();
        reproductor = MediaPlayer.create(this, R.raw.jack);
        reproductor.setLooping(false);
        reproductor.setVolume(100, 100);
    }

    public int onStartCommand(Intent intent, int flags, int startId) {

        reproductor.start();

        return START_STICKY;
    }

    public void onDestroy() {
        super.onDestroy();
        if (reproductor.isPlaying()) {
            reproductor.stop();
        }
        reproductor.release();
        reproductor = null;
    }

    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}
```



JAVA:

```
package com.game.magody.fragments;

import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

/**
 * A simple {@link Fragment} subclass.
 */
public class MusicaFragment extends Fragment {

    private boolean encendida;
    private ImageView boton_musica;

    public MusicaFragment() {
        // Required empty public constructor
    }

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View fragmento = inflater.inflate(R.layout.fragment_musica,
        container, false);

        boton_musica = (ImageView) fragmento.findViewById(R.id.musica);

        if(encendida)
            boton_musica.setImageResource(R.drawable.musica2);

        getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_POR
        TRAIT);

        boton_musica.setOnClickListener(new View.OnClickListener() {
```

```

        public void onClick(View vista){

            if(encendida)
                apagaMusica();
            else
                enciendeMusica();

            encendida = !encendida;

        }

    });

    return fragmento;
}

private void enciendeMusica(){

    boton_musica.setImageResource(R.drawable.musica2);
    Intent intent = new Intent(getActivity(), ServicioMusica.class);
    getActivity().startService(intent);

}

private void apagaMusica(){

    boton_musica.setImageResource(R.drawable.musica);
    Intent intent = new Intent(getActivity(), ServicioMusica.class);
    getActivity().stopService(intent);

}

/*
public void onPause(){
    super.onPause();
    Intent intent = new Intent(getActivity(), ServicioMusica.class);
    getActivity().stopService(intent);
}*/

}

```

XML:

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:gravity="center"
tools:context="com.game.magody.fragments.MusicaFragment">

<ImageView
    android:id="@+id/musica"
    android:layout_width="@dimen/maximo"

```

```
        android:layout_height="@dimen/maximo"  
        android:src="@drawable/musica" />
```

```
</RelativeLayout>
```

Uso:

```
fragments = new Fragment[3];  
fragments[2] = new MusicaFragment();
```

Fragmento de acelerómetro (nivelador)

Interfaz:

Clases

DIBUJOS DEL ACELEROMETRO

```
package com.game.magody.fragments;
```

```
import android.content.Context;
```

```
import android.graphics.Bitmap;
```

```
import android.graphics.Canvas;
```

```
import android.graphics.Color;
```

```
import android.graphics.Paint;
```

```
import android.graphics.drawable.BitmapDrawable;
```

```
import android.support.v4.content.ContextCompat;
```

```
import android.support.v7.widget.AppCompatImageView;
```

```
import android.widget.ImageView;
```

```
import java.text.DecimalFormat;
```

```
/**
 * Created by Juan on 09/05/2017.
 */

public class NivelPantalla extends AppCompatActivity {

    int lado, radio, radioPeq, trazo;

    float[] angulos; // Recibirá las variaciones del sensor

    Bitmap fondo; // dibujo de fondo

    Paint trazoDibujo;

    Bitmap burbuja;

    public NivelPantalla(Context contexto, int lado){ // recibe el tamaño del espacio de dibujo con
"lado"

        super(contexto);

        this.lado=lado;

        radio=lado/2;

        radioPeq=lado/10;

        trazo=lado/100;
```

```
angulos=new float[2];
```

```
angulos[0]=0;
```

```
angulos[1]=0;
```

```
fondo=inciaFondo();
```

```
trazoDibujo=new Paint();
```

```
trazoDibujo.setColor(Color.BLACK);
```

```
trazoDibujo.setTextSize(20);
```

```
BitmapDrawable bola=(BitmapDrawable) ContextCompat.getDrawable(contexto,  
R.drawable.burbuja);
```

```
burbuja=bola.getBitmap();
```

```
burbuja=Bitmap.createScaledBitmap(burbuja, radioPeq*2, radioPeq*2, true);
```

```
}
```

```
private Bitmap inciaFondo(){
```

```
Bitmap.Config conf=Bitmap.Config.ARGB_4444;
```

```
Bitmap fondo=Bitmap.createBitmap(lado, lado, conf);
```

```
Canvas lienzo=new Canvas(fondo);
```

```
Paint lapiz=new Paint();
```

```
lapiz.setColor(Color.RED);
```

```
lienzo.drawCircle(radio, radio, radio, lapiz);
```

```
lapiz.setColor(Color.BLACK);
```

```
lienzo.drawCircle(radio, radio, radio-trazo, lapiz);
```

```
lapiz.setColor(Color.RED);
```

```
lienzo.drawCircle(radio, radio, radioPeq+trazo, lapiz);
```

```
lapiz.setStrokeWidth(trazo);
```

```
lienzo.drawLine(radio, 0, radio, lado, lapiz);
```

```
lienzo.drawLine(0, radio, lado, radio, lapiz);
```

```
return fondo;
```

```
}
```

```
public void angulos(float[] angulos){
```

```
this.angulos=angulos;  
  
// redibuja  
invalidate();  
  
}
```

protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec){ // ajustamos  
dimensiones de la vista (no trabaja a pantalla completa)

```
super.onMeasure(widthMeasureSpec, heightMeasureSpec);  
  
setMeasuredDimension(lado, lado);  
  
}
```

protected void onDraw(Canvas lienzo){

```
super.onDraw(lienzo);
```

```
lienzo.drawBitmap(fondo, 0, 0, null);
```

```
int posX=radio-radioPeq+(int)(angulos[0]/10*radio);
```

```
int posY=radio-radioPeq-(int)(angulos[1]/10*radio);
```

```
lienzo.drawBitmap(burbuja, posX, posY, null);
```

```
}  
  
}
```

JAVA:

```
package com.game.magody.fragments;  
  
import android.content.Context;  
import android.content.pm.ActivityInfo;  
import android.hardware.Sensor;  
import android.hardware.SensorEvent;  
import android.hardware.SensorEventListener;  
import android.hardware.SensorManager;  
import android.os.Bundle;  
import android.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
/**  
 * A simple {@link Fragment} subclass.  
 */  
public class NivelFragment extends Fragment implements  
    SensorEventListener {  
  
    private SensorManager manager;  
    private Sensor sensor;  
  
    private NivelPantalla pantalla;  
  
    public NivelFragment() {  
        // Required empty public constructor  
    }  
  
    public final void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        manager = (SensorManager)  
            getActivity().getSystemService(Context.SENSOR_SERVICE);  
        sensor = manager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);  
  
        int lado = getResources().getDimensionPixelSize(R.dimen.maximo);  
        pantalla = new NivelPantalla(getActivity(), lado);  
    }  
}
```



```

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        /*return inflater.inflate(R.layout.fragment nivel, container,
false);*/

        // IMPORTANTE CAMBIO

        getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_POR
TRAIT);
        return pantalla;
    }

    public void onResume() {

        super.onResume();

        manager.registerListener(this, sensor, SensorManager.SENSOR_DELAY_GAME);
    }

    public void onPause() {
        super.onPause();
        manager.unregisterListener(this);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {

        pantalla.angulos(event.values);
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }

}

```

XML:

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"

```

```

        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical"
        tools:context="com.game.magody.fragments.NivelFragment">

        <ImageView
            android:id="@+id/nivel"
            android:layout_width="@dimen/maximo"
            android:layout_height="@dimen/maximo"
            android:src="@drawable/nivel" />

    </RelativeLayout>

```

Uso:

```

fragments = new Fragment[3];
fragments[1] = new NivelFragment();

```

Fragmento de linterna

```

public interface ManejaFlashCamara {

    void enciendeApaga(boolean estado_camara);

}

public class HerramientasActivity extends AppCompatActivity implements
ComunicaMenu, ManejaFlashCamara {

    private CameraManager camara;
    private String id_camara;

```

Dentro de onCreate de la clase invocadora

```

camara = (CameraManager) getSystemService(Context.CAMERA_SERVICE);

try{

    id_camara = camara.getCameraIdList()[0];
}catch (Exception e){
    e.printStackTrace();
}

```

Y se sobrescribe

```

@Override
public void enciendeApaga(boolean estado_camara) {
    Toast toast;
    String salida = "Flash ";

```

```

    try{
        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) // marshmallow
6.9 mayor a esa
            camara.setTorchMode(id_camara,!estado_camara);

    }catch (Exception e){
        e.printStackTrace();
    }

    salida += (estado_camara)? "apagado": "activado";

    toast = Toast.makeText(this,salida, Toast.LENGTH_SHORT);
    toast.show();

}

```

## Permisos

```

<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.FLASHLIGHT"/>

<uses-feature android:name="android.hardware.camera"/>
<uses-feature android:name="android.hardware.camera.flash"/>

```

## JAVA:

```

package com.game.magody.fragments;

import android.app.Activity;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

/**
 * A simple {@link Fragment} subclass.
 */
public class LinternaFragment extends Fragment {

    private ImageView boton_camara;
    private boolean encendida;

    public LinternaFragment() {
        // Required empty public constructor
    }

```

```

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                               Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View fragmento = inflater.inflate(R.layout.fragment_linterna,
container, false);

        // modificamos una parte del fragmento
        boton_camara = (ImageView) fragmento.findViewById(R.id.linterna);

        if(encendida){
            botonEnciendeFlash();
        }

        getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_POR
TRAIT);

        boton_camara.setOnClickListener(new View.OnClickListener(){

            public void onClick(View v){

                if(encendida){
                    botonApagaFlash();
                }
                else
                {
                    botonEnciendeFlash();
                }
                Activity esta_actividad = getActivity();

                // casting a la interface

                ((ManejaFlashCamara)esta_actividad).enciendeApaga(encendida);

                encendida = !encendida;

            }

        });

        return fragmento;
    }

    public void botonEnciendeFlash(){
        boton_camara.setImageResource(R.drawable.linterna2);
    }

```

```

    }

    public void botonApagaFlash() {
        boton_camara.setImageResource(R.drawable.linterna);
    }
}

```

XML:

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:gravity="center" <!--para ubicar bien la imagen-->
tools:context="com.game.magody.fragments.LinternaFragment">

    <ImageView
        android:id="@+id/linterna"
        android:layout_width="@dimen/maximo"
        android:layout_height="@dimen/maximo"
        android:src="@drawable/linterna" />

</RelativeLayout>

```

Uso:

```

fragments = new Fragment[3];
fragments[0] = new LinternaFragment();

```

## Firestore

Reglas para usar los datos

Autenticación -> rules false false

Pruebas -> rules true true

si sale un error conectamos de nuevo sincronizándonos

## signOut

```

public void signOut(View view){

    firebaseAuth.signOut();

    updateUI(null);

}

```

## Email verification

```
private void sendEmailVerification() {  
  
    // Disable button  
  
    findViewById(R.id.verifyEmailButton).setEnabled(false);  
  
  
    // Send verification email  
  
    // [START send_email_verification]  
  
    final FirebaseUser user = mAuth.getCurrentUser();  
  
    user.sendEmailVerification()  
  
        .addOnCompleteListener(this, new OnCompleteListener<Void>() {  
  
            @Override  
  
            public void onComplete(@NonNull Task<Void> task) {  
  
                // [START_EXCLUDE]  
  
                // Re-enable button  
  
                findViewById(R.id.verifyEmailButton).setEnabled(true);  
  
  
                if (task.isSuccessful()) {  
  
                    Toast.makeText(EmailPasswordActivity.this,  
                        "Verification email sent to " + user.getEmail(),  
                        Toast.LENGTH_SHORT).show();  
  
                } else {  
  
                    Log.e(TAG, "sendEmailVerification", task.getException());  
  
                    Toast.makeText(EmailPasswordActivity.this,  
                        "Failed to send verification email.",  
                        Toast.LENGTH_SHORT).show();  
  
                }  
  
                // [END_EXCLUDE]  
  
            }  
  
        });  
};
```

```
// [END send_email_verification]
```

```
}
```

## Registrar

```
private DatabaseReference clasesDatabase;
```

```
private final static String[] asignaturas = { "Asignatura A", "Asignatura B", "Asignatura C",  
        "Asignatura D", "Asignatura E", "Asignatura F" };
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_crud);
```

```
    Toolbar toolbar = findViewById(R.id.toolbar);
```

```
    setSupportActionBar(toolbar);
```

```
    clasesDatabase = FirebaseDatabase.getInstance().getReference("Clases");
```

```
    editTextTema = (EditText) findViewById(R.id.editTextTema);
```

```
    spinnerSeccion = (Spinner) findViewById(R.id.spinnerSeccion);
```

```
    spinnerAsignatura = (Spinner) findViewById(R.id.spinnerAsignatura);
```

```
    buttonRegistrar = (Button) findViewById(R.id.buttonRegistrar);
```

```
    ArrayAdapter adapter = new ArrayAdapter<String>(this,
```

```
        android.R.layout.simple_spinner_dropdown_item, asignaturas);
```

```
    spinnerAsignatura.setAdapter(adapter);
```

```
}
```

```

public void registrarClase(View view){

    String seccion = spinnerSeccion.getSelectedItem().toString();

    String asignatura = spinnerAsignatura.getSelectedItem().toString();

    String tema = editTextTema.getText().toString();


    if(!TextUtils.isEmpty(tema)){

        String id = clasesDatabase.push().getKey();

        Clases leccion = new Clases(id, seccion, asignatura, tema);

        clasesDatabase.child("lecciones").child(id).setValue(leccion);

        Toast.makeText(this, "Clase registrada", Toast.LENGTH_LONG).show();

    }else{

        Toast.makeText(this, "Tema requerido", Toast.LENGTH_LONG).show();

    }

}

FirebaseUser user = firebaseAuth.getCurrentUser();


        DatabaseReference databaseReferenceUber =
        FirebaseDatabase.getInstance().getReference("Usuarios");

databaseReferenceUber.child("Clientes").child(user.getUid()).child("email").setValue(user.getEmail());

databaseReferenceUber.child("Clientes").child(user.getUid()).child("Nombre").setValue("Desconocido123");


        //POR CLASES O NORMALClases leccion = new Clases(id, seccion, asignatura,
tema);

        //      clasesDatabase.child("lecciones").child(id).setValue(leccion);

```



## Geofire

```
package com.app.magody.appuber;
```

```
import androidx.annotation.NonNull;
```

```
import androidx.annotation.Nullable;
```

```
import androidx.core.app.ActivityCompat;
```

```
import androidx.fragment.app.FragmentActivity;
```

```
import android.Manifest;
```

```
import android.content.pm.PackageManager;
```

```
import android.location.Location;
```

```
import android.os.Bundle;
```

```
import android.util.Log;
```

```
import android.widget.Toast;
```

```
import com.firebase.geofire.GeoFire;
```

```
import com.google.android.gms.common.ConnectionResult;
```

```
import com.google.android.gms.common.api.GoogleApiClient;
```

```
import com.google.android.gms.location.LocationRequest;
```

```
import com.google.android.gms.location.LocationServices;
```

```
import com.google.android.gms.maps.CameraUpdateFactory;
```

```
import com.google.android.gms.maps.GoogleMap;
```

```
import com.google.android.gms.maps.OnMapReadyCallback;
```

```
import com.google.android.gms.maps.SupportMapFragment;
```

```
import com.google.android.gms.maps.model.LatLng;
```

```
import com.google.firebase.auth.FirebaseAuth;
```

```
import com.google.firebase.database.DatabaseReference;
```

```
import com.google.firebase.database.FirebaseDatabase;
```

```
public class MapsClienteActivity extends FragmentActivity implements OnMapReadyCallback,  
GoogleApiClient.ConnectionCallbacks,
```

```
    GoogleApiClient.OnConnectionFailedListener,  
    com.google.android.gms.location.LocationListener {
```

```
    private GoogleMap mMap;
```

```
    GoogleApiClient mGoogleApiClient;
```

```
    Location mLastLocation;
```

```
    LocationRequest mLocationRequest;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_maps_cliente);
```

```
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
```

```
        SupportMapFragment mapFragment = (SupportMapFragment)  
        getSupportFragmentManager()
```

```
            .findFragmentById(R.id.map);
```

```
        if (ActivityCompat.checkSelfPermission(this,  
        android.Manifest.permission.ACCESS_FINE_LOCATION) !=  
        PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,  
        android.Manifest.permission.ACCESS_COARSE_LOCATION) !=  
        PackageManager.PERMISSION_GRANTED) {
```

```
            ActivityCompat.requestPermissions(MapsClienteActivity.this, new  
            String[]{Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_REQUEST); // se asocia a ese  
            código
```

```

    }else{
        try {
            mapFragment.getMapAsync(this); // empieza el mapa
        }catch (Exception ex){
            Log.d("ERROR", ex.getMessage());
        }
    }
}

/**
 * Manipulates the map once available.
 * This callback is triggered when the map is ready to be used.
 * This is where we can add markers or lines, add listeners or move the camera. In this case,
 * we just add a marker near Sydney, Australia.
 * If Google Play services is not installed on the device, the user will be prompted to install
 * it inside the SupportMapFragment. This method will only be triggered once the user has
 * installed Google Play services and returned to the app.
 */
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    // Add a marker in Sydney and move the camera
    //LatLng sydney = new LatLng(-34, 151);
    //mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));

```

```
//mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
```

```
if (ActivityCompat.checkSelfPermission(this,  
    android.Manifest.permission.ACCESS_FINE_LOCATION) !=  
    PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,  
    android.Manifest.permission.ACCESS_COARSE_LOCATION) !=  
    PackageManager.PERMISSION_GRANTED) {
```

```
    ActivityCompat.requestPermissions(MapsClienteActivity.this, new  
    String[]{Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_REQUEST); // se asocia a ese  
    código
```

```
}
```

```
    buildGoogleApiClient();
```

```
    mMap.setMyLocationEnabled(true);
```

```
}
```

```
protected synchronized void buildGoogleApiClient(){  
    mGoogleApiClient = new GoogleApiClient.Builder(this)  
        .addConnectionCallbacks(this)  
        .addOnConnectionFailedListener(this)  
        .addApi(LocationServices.API)  
        .build();  
    mGoogleApiClient.connect();  
}
```

```
@Override
```

```
public void onLocationChanged(Location location) {
```

```

if(getApplicationContext()!=null){
    mLastLocation = location;

    LatLng latLng = new LatLng(location.getLatitude(),location.getLongitude());

    mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
    mMap.animateCamera(CameraUpdateFactory.zoomTo(11));// el más cercano es ... tipos de
zoom 1-21

```

```

    /*String userId = FirebaseAuth.getInstance().getCurrentUser().getUid();

    DatabaseReference ref =
    FirebaseDatabase.getInstance().getReference("TrabajadoresDisponibles");

    GeoFire geoFire = new GeoFire(ref);

    geoFire.setLocation(userId, new GeoLocation(location.getLatitude(),
location.getLongitude()));*/
}

}

```

```

@Override
public void onConnected(@Nullable Bundle bundle) {
    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(1000);
    mLocationRequest.setFastestInterval(1000);
    mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY); //HIGH usa
mucha batería

```

```

        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(MapsClienteActivity.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_REQUEST); // se asocia a ese
código

        }

        LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,
mLocationRequest, this);

    }

    @Override

    public void onConnectionSuspended(int i) {

    }

    @Override

    public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

    }

    @Override

    public void onPointerCaptureChanged(boolean hasCapture) {

    }

    @Override

```

```

protected void onStop() {

    super.onStop();

    /*

    LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient, this);

    String userId = FirebaseAuth.getInstance().getCurrentUser().getUid();

    DatabaseReference ref =
    FirebaseDatabase.getInstance().getReference("TrabajadoresDisponibles");

    GeoFire geoFire = new GeoFire(ref);

    geoFire.removeLocation(userId);*/

}

final int LOCATION_REQUEST = 1;

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {

    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    switch (requestCode){

        case LOCATION_REQUEST:

            if(grantResults.length > 0){

                if(grantResults[0] == PackageManager.PERMISSION_GRANTED){

                    SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()

                        .findFragmentById(R.id.map);

                    mapFragment.getMapAsync(this); // empieza el mapa

                }else{

                    print(getString(R.string.mensajePermisoGPS));

                }

            }

        }

    }
}

```

```

        break;
    }

}

private void print(String texto){
    Toast.makeText(this,texto,Toast.LENGTH_LONG).show();
}
}

```

## Bases de datos

Create

Read

Update

Delete

## Juegos

Set content view – establecer una vista desde un programa

```
partida = new Partida(this,dificultad);
```

```
setContentView(partida);
```

## Dibujo del juego

```

package com.game.magody.hilospersistenciasonido;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Point;
import android.graphics.drawable.BitmapDrawable;
import android.support.v4.content.ContextCompat;
import android.view.Display;
import android.view.WindowManager;
import android.support.v7.widget.AppCompatImageView;

public class Partida extends AppCompatImageView {

```



```

private int acel;
private Bitmap pelota, fondo;
private int tam_pantX, tam_pantY, posX, posY, velX, velY;
private int tamPelota;
boolean pelota_sube;

public Partida(Context contexto, int nivel_dificultad){
    // INFO en la API de Android
    super(contexto);

    // rescata datos del display del dispositivo
    WindowManager manejador_ventana=(WindowManager)
contexto.getSystemService(Context.WINDOW_SERVICE);
    Display pantalla=manejador_ventana.getDefaultDisplay();

    // integra dos coordenadas x,y
    Point maneja_coord=new Point();

    pantalla.getSize(maneja_coord); // determina el tamaño de la
pantalla: horizontal y vertical

    tam_pantX=maneja_coord.x; // obtiene el horizontal tamaño

    tam_pantY=maneja_coord.y; // obtiene el vertical tamaño

    // Construye un Layout programatico, interfaz mediante codigo
    // contextCompat permite acceder a características del contexto
    BitmapDrawable dibujo_fondo = (BitmapDrawable)
ContextCompat.getDrawable(contexto, R.drawable.paisaje_1);

    fondo=dibujo_fondo.getBitmap();// mirar en api getBitmap en clase
BitmapDrawable. esto nos lleva a la siguiente instr.

    // aplica un escalado al fondo
    fondo=Bitmap.createScaledBitmap(fondo, tam_pantX, tam_pantY,
false);//mirar en clase Bitmap

    // lo mismo para la pelota
    BitmapDrawable
objetoPelota=(BitmapDrawable)ContextCompat.getDrawable(contexto,
R.drawable.pelota_2);

    pelota=objetoPelota.getBitmap();

    // determina la dificultad
    tamPelota=tam_pantY/3;

    pelota=Bitmap.createScaledBitmap(pelota, tamPelota, tamPelota,
false);

    // en el centro horizontal
    posX=tam_pantX/2-tamPelota/2;
    // mas arriba de la pantalla
    posY=0-tamPelota;

```

```

        // aceleración dada por la dificultad
        acel=nivel_dificultad*(maneja_coord.y/400);

    }

    public boolean toque(int x, int y){
        // cuando se toca la pantalla, recibe el lugar donde toco con el
        dedo

        //invalida el toque en la primera tercera parte del eje y
        if(y<tam_pantY/3)
            return false;

        //si la velocidad en y es menor o igual a 0, no debe funcionar el
        metodo
        if(velY<=0)
            return false;

        // si no acertamos al tocar la pelota, no hace nada
        if(x<posX || x>posX+tamPelota)
            return false;

        if(y<posY || y>posY+tamPelota)
            return false;

        // invierte la velocidad en y
        velY=-velY;

        // impulso en X
        double desplX=x-(posX+tamPelota/2);

        desplX=desplX/(tamPelota/2)*velY/2;

        velX+=(int)desplX;

        return true;
    }

    public boolean movimientoBola(){
        // devuelve true cuando el juego ha terminado

        if(posX<0-tamPelota){
            // se va a la izquierda y se repone arriba
            posY=0-tamPelota;

            velY=acel;
        }

        posX+=velX;

        posY+=velY;

        // cuando topa el suelo
        if(posY>=tam_pantY)
            return true;
    }

```

```

        //se fue a la izquierda o derecha pierde
        if(posX+tamPelota<0 || posX>tam_pantX)
            return true;

        if(velY<0)
            pelota_sube=true;

        if(velY>0 && pelota_sube){

            pelota_sube=false;

        }

        velY+=acel;

        return false;
    }

    protected void onDraw(Canvas lienzo){

        lienzo.drawBitmap(fondo, 0,0, null);

        lienzo.drawBitmap(pelota, posX, posY, null);

    }
}

```

## Carpeta RES

Los nuevos recursos se deben agregar usando el GUI encargado de ello

### Drawable

Aquí se guardan las imágenes en varios tamaños

Crear Objetos reutilizables

fondo.xml

```

<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item>
        <shape>
            <!-- shape es el fondo base-->

            <gradient
                android:startColor="#0095ff"
                android:endColor="#4cffba"

```

```

        android:angle="0"/>

    </shape>

</item>

</selector>

```

boton.xml

```

<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:state_pressed="true">
        <shape>
            <solid android:color="#f98ec0"/>
            <corners android:radius="@dimen/radio_boton"/>
        </shape>
    </item>

    <item android:state_enabled="false">
        <shape>
            <solid android:color="#303f8f"/>
            <corners android:radius="@dimen/radio_boton"/>
        </shape>
    </item>

    <item>
        <shape>
            <gradient
                android:startColor="#91ff2a"
                android:endColor="#25a344"
                android:angle="90"/>
            <corners android:radius="@dimen/radio_boton"/>
        </shape>
    </item>

</selector>

```

Y en styles:

```

<style name="boton">
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:minWidth">@dimen/ancho_boton</item>
    <item name="android:minHeight">@dimen/alto_boton</item>
    <item name="android:layout_margin">@dimen/margen_boton</item>
    <item name="android:textSize">@dimen/texto_boton</item>
    <item name="android:onClick">jugar</item>
    <item name="android:background">@drawable/boton</item>

```

```

        <item name="android:textColor">#000</item>
    </style>

```

## Layout

Aquí se almacenan los XML de las actividades correspondientes, es decir la interfaz gráfica:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"

    tools:context="com.game.magody.tablasdinamicas.MainActivity">

    <Button
        android:id="@+id/btn_reset"
        android:layout_width="372dp"
        android:layout_height="wrap_content"
        android:onClick="reset"
        android:text="@string/btn_reset"
        android:textAllCaps="false"
        android:textStyle="bold|italic"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" />

    <TextView
        android:id="@+id/tv_archivo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <ScrollView
        android:layout_width="368dp"
        android:layout_height="305dp"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        tools:layout_editor_absoluteY="239dp">

        <HorizontalScrollView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fillViewport="true"><!-- el fill view port rellena el
horizontal de la tabla-->

            <TableLayout
                android:id="@+id/table"
                android:layout_width="wrap_content"
                android:layout_height="match_parent">

```

```
</TableLayout>
</HorizontalScrollView>
```

```
</ScrollView>
```

```
<android.support.constraint.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button"
        android:layout_width="94dp"
        android:layout_height="39dp"
        android:layout_marginEnd="8dp"
        android:onClick="verHorario"
        android:text="@string/btn_horario"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.514"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.884" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginTop="32dp"
        android:onClick="mostrarNotas"
        android:text="@string/btn_notas"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0" />

    <Button
        android:id="@+id/btn_salir"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="16dp"
        android:layout_marginStart="6dp"
        android:onClick="salir"
        android:text="@string/btn_salir"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.971"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.875" />
```

```

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab_agregar"
    android:layout_width="72dp"
    android:layout_height="64dp"
    android:layout_gravity="end|bottom"
    android:layout_marginStart="8dp"
    android:layout_weight="1"
    android:clickable="true"
    android:focusable="true"
    android:src="@android:drawable/ic_input_add"
    app:backgroundTint="@color/colorFloating"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.916"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.21" />

<Button
    android:id="@+id/button7"
    android:layout_width="wrap_content"
    android:layout_height="91dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:onClick="verImagen"
    android:text="Imagen"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.11"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.897" />
</android.support.constraint.ConstraintLayout>

```

```
</LinearLayout>
```

## MipMap

Es un lugar que almacena los íconos de la aplicación

## Values

La carpeta values puede guardar las strings del idioma, estilos y color

## colors.xml

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

```
<resources>

    <color name="colorPrimary">#3F51B5</color>

    <color name="colorPrimaryDark">#303F9F</color>

    <color name="colorAccent">#FF4081</color>

</resources>
```

#### strings.xml

```
<resources>

    <string name="app_name">Asteroides</string>
    <string name="titulo">¡Asteroides!</string>

    <string name="btnjugar">¡Jugar!</string>
    <string name="btnconfigurar">Configuración</string>
    <string name="btnsalir">Salir</string>
    <string name="btninfo">Información</string>

    <!--Informacion.xml-->

    <string name="informacion_titulo">Información importante</string>

    <string name="informacion_desc">Danny Diaz.\n20 años.\nEscuela Politecnica
nacional\nIng.Sistemas</string>

</resources>
```

#### styles.xml

Estilo 1:

```
<resources>

    <string name="app_name">Asteroides</string>
    <string name="titulo">¡Asteroides!</string>

    <string name="btnjugar">¡Jugar!</string>
    <string name="btnconfigurar">Configuración</string>
```



```

<string name="btnsalir">Salir</string>

<string name="btninfo">Información</string>

<!--Informacion.xml-->

<string name="informacion_titulo">Información importante</string>

<string name="informacion_desc">Danny Diaz.\n20 años.\nEscuela Politecnica
nacional\nIng.Sistemas</string>

</resources>

```

Estilo 2:

```

<resources>

<!-- Base application theme. -->

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

    <!-- Customize your theme here. -->

    <item name="colorPrimary">@color/colorPrimary</item>

    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>

    <item name="colorAccent">@color/colorAccent</item>

</style>

</resources>

```

Estilo con botones

```

<style name="botones">
    <item name="android:layout_width">@dimen/largo</item>
    <item name="android:layout_height">@dimen/largo</item>

    <!--escalado desde el centro, para que aparezca toda la imagen-->
    <item name="android:scaleType">fitCenter</item>

</style>

```

Estilo de un espacio

```

<style name="espacio">
    <item name="android:layout_width">30dp</item>
    <item name="android:layout_height">1dp</item>
</style>

```

*Style personalizado con mecánicas*

```

<style name="casilla">

    <item name="android:layout_width">@dimen/casilla</item>

    <item name="android:layout_height">@dimen/casilla</item>

    <item name="android:src">@drawable/casilla</item>

    <item name="android:onClick">toque</item>

    <item name="android:layout_margin">4dp</item>

</style>

```

dimens.xml

Al igual que strings permite sobrecarga (para normal o large, etc). Almacena las dimensiones

Para normal:

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="minimo">16dp</dimen>
    <dimen name="medio">32dp</dimen>
    <dimen name="largo">60dp</dimen>
    <dimen name="maximo">300dp</dimen>

    <dimen name="texto">12dp</dimen>

</resources>

```

Para Large:

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="minimo">32dp</dimen>
    <dimen name="medio">64dp</dimen>
    <dimen name="largo">120dp</dimen>
    <dimen name="maximo">600dp</dimen>

    <dimen name="texto">24dp</dimen>

```

```
</resources>
```

Raw

Aquí pueden guardarse archivos mp3

APIs

Google Maps

```
<string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">AlzaSyDxAZPiQkCiuy3hyHnY9V856sYqTiuGVX8</string>
```

Clases

Tablas

```
package com.game.magody.tablasdinamicas.tablas;
```

```
import android.content.Context;
```

```
import android.view.Gravity;
```

```
import android.view.View;
```

```
import android.widget.TableLayout;
```

```
import android.widget.TableRow;
```

```
import android.widget.TextView;
```

```
import java.util.ArrayList;
```

```
public abstract class TableDinamic implements View.OnClickListener{
```

```
    protected TableLayout tableLayout;
```

```
    protected Context context;
```

```
    protected String[] header;
```

```

protected ArrayList<String[]> datos;

protected boolean multiColor= false;

protected int first_color, second_color, text_color;


public static String meta,seccion;


public TableDinamic(TableLayout tableLayout, Context context){

    this.tableLayout = tableLayout;

    this.context = context;

}


public void crearHeader(String[] header) {

    this.header = header;

    TableRow tabla_fila = new TableRow(context);

    for (String h : header) { //total de datos dentro del arreglo

        TextView tv_celda = new TextView(context);

        tv_celda.setGravity(Gravity.CENTER);

        tv_celda.setTextSize(25);

        tv_celda.setText(h);

        tabla_fila.addView(tv_celda, parametrosFila());

    }

    tableLayout.addView(tabla_fila);

}


public abstract void agregarFilasData(ArrayList<String[]> data);


protected TableRow.LayoutParams parametrosFila(){

    TableRow.LayoutParams params = new TableRow.LayoutParams();

    params.setMargins(1,1,1,1); // separacion de celdas

```

```

        params.weight = 1; // de forma proporcional

        return params;

    }

    public void reColoring(){

        multiColor = !multiColor;

        for(int indexC=0;indexC<header.length;indexC++){ //total de datos dentro del arreglo

            TextView tv_celda = getCelda(datos.size(),indexC);

            tv_celda.setBackgroundColor((multiColor)? first_color:second_color);

            tv_celda.setTextColor(text_color);

        }

    }

    protected TextView getCelda(int row_index, int column_index){

        TableRow tabla_fila = getFila(row_index);

        return (TextView) tabla_fila.getChildAt(column_index);

    }

    protected TableRow getFila(int index){

        return (TableRow) tableLayout.getChildAt(index);

    }

    public void backgroundHeader(int color){

        for(int indexC=0;indexC<header.length;indexC++){ //total de datos dentro del arreglo

            TextView tv_celda = getCelda(0,indexC);

            tv_celda.setBackgroundColor(color);

```

```
}  
}
```

```
public void backgroundData(int first_color,int second_color){  
  
    for(int indexR = 1; indexR<= datos.size(); indexR++){  
        //desde la 1 porque en la cero esta el encabezado  
        multiColor = !multiColor; // lineas de diferentes colores  
        for(int indexC=0;indexC<header.length;indexC++){  
            TextView tv_celda = getCelda(indexR,indexC);  
            tv_celda.setBackgroundColor((multiColor)? first_color:second_color);  
        }  
    }  
  
    this.first_color = first_color;  
    this.second_color = second_color;  
  
}
```

```
public void lineColor(int color){  
  
    for(int indexR = 0; indexR< datos.size(); indexR++){  
        getFila(indexR).setBackgroundColor(color);  
    }  
}
```

```
public void textColorHeader(int color){
```

```
        for(int indexC=0;indexC<header.length;indexC++){  
            getCelda(0,indexC).setTextColor(color);  
        }  
    }
```

```
public void textColorData(int color){  
    for(int indexR = 1; indexR<= datos.size(); indexR++){  
        for(int indexC=0;indexC<header.length;indexC++){  
            getCelda(indexR,indexC).setTextColor(color);  
        }  
        this.text_color = color;  
    }  
}
```

```
}
```

```
package com.game.magody.tablasdinamicas.tablas;  
  
import android.content.Context;  
import android.view.Gravity;  
import android.view.View;  
import android.widget.TableLayout;  
import android.widget.TableRow;  
import android.widget.TextView;  
import com.game.magody.tablasdinamicas.MainActivity;
```

```

import com.game.magody.tablasdinamicas.Modelos.Meta;

import com.game.magody.tablasdinamicas.herramientas.ComparadorTitulo;


import java.util.ArrayList;
import java.util.Collections;


public class TablaMetas extends TableDinamic {


    private MainActivity actividad_main;


    public TablaMetas(TableLayout tableLayout, Context context, MainActivity actividad_actual) {
        super(tableLayout, context);


        this.actividad_main = actividad_actual;
    }


    public void agregarFilasData(ArrayList<String[]> data) {
        this.datos = data;

        String info;

        for (int indexR = 1; indexR <= data.size(); indexR++) {
            //desde la 1 porque en la cero esta el encabezado

            TableRow tabla_fila = new TableRow(context);
            tabla_fila.setOnLongClickListener(new View.OnLongClickListener() {
                @Override
                public boolean onLongClick(View v) {
                    meta = ((TextView) ((TableRow) v).getChildAt(0)).getText().toString();

                    actividad_main.alerta_eliminar.show();

                    return true;
                }
            });
            tabla_fila.addView(meta);
            tableLayout.addView(tabla_fila);
        }
    }
}

```



```

    }
    });
    tabla_fila.setOnClickListener(this);
    for (int indexC = 0; indexC < header.length; indexC++) {
        TextView tv_celda = new TextView(context);
        tv_celda.setGravity(Gravity.CENTER);
        tv_celda.setTextSize(25);
        String[] row = data.get(indexR - 1); // en la posicion cero estará la informacion
        info = (indexC < row.length) ? row[indexC] : ""; // si solo tengo tres columnas en las
        demas se agrega un vacio
        tv_celda.setText(info);
        tabla_fila.addView(tv_celda, parametrosFila());
    }
    tableLayout.addView(tabla_fila);

}
}

```

```

public void agregarFilaInfo(String[] item){

```

```

    String info;

```

```

    datos.add(item);

```

```

    TableRow tabla_fila = new TableRow(context);

```

```

    tabla_fila.setOnLongClickListener(new View.OnLongClickListener() {

```

```

        @Override

```

```

        public boolean onLongClick(View v) {

```

```

            meta = ((TextView) ((TableRow) v).getChildAt(0)).getText().toString();

```

```

            actividad_main.alerta_eliminar.show();

```

```

            return true;
        }
    }
}

```

```

    }
});
tabla_fila.setOnClickListener(this);
for(int indexC = 0;indexC<header.length; indexC++){
    TextView tv_celda = new TextView(context);
    tv_celda.setGravity(Gravity.CENTER);
    tv_celda.setTextSize(25);
    info = (indexC<item.length)? item[indexC]:""; // si solo tengo tres columnas en las demas se
    agrega un vacio
    tv_celda.setText(info);
    tabla_fila.addView(tv_celda, parametrosFila());
}
tableLayout.addView(tabla_fila, datos.size()); // inserta en la ultima posicion

reColoring();
}

```

```

@Override
public void onClick(View v) {

    Meta aux = new Meta(((TextView) ((TableRow) v).getChildAt(0)).getText().toString());
    boolean es_presicion = false;

    int index = Collections.binarySearch(MainActivity.metas_objetivos, aux, new
    ComparadorTitulo());

```

```

        if (index < 0) {

            index = Collections.binarySearch(MainActivity.metas_presicion, aux, new
ComparadorTitulo());

            es_presicion = true;

        }

        if (index >= 0) {

            if (es_presicion) {

                actividad_main.pasosActivity(MainActivity.metas_presicion.get(index),
es_presicion,index);

            } else {

                actividad_main.pasosActivity(MainActivity.metas_objetivos.get(index),
es_presicion,index);

            }

        }

    }
}

```

## Fecha

```
package com.game.magody.tablasdinamicas.Modelos;
```

```
import java.util.Calendar;
```

```
import java.util.GregorianCalendar;
```

```
import java.util.StringTokenizer;
```

```
import java.util.regex.Matcher;
```

```
import java.util.regex.Pattern;
```

```
/**
```

\* Created by User on 09/08/2018.

\*/

```
public class Fecha {

    //Formato de fecha dia/mes/año, - o /

    public int dia, mes, año;

    public Fecha() {

    }

    public Fecha(int day, int month, int year) {

        super();

        this.dia = day;

        this.mes = month;

        this.año = year;

    }

    public boolean establecerFecha(String fecha) {

        Pattern patron_fecha = Pattern.compile("^([0-9]|[0-2][0-9]|3[0-1])/([0-9]|1[0-2])/([0-9]){4}$");

        Matcher coincidencia;

        coincidencia = patron_fecha.matcher(fecha);

        if (coincidencia.matches()) {

            StringTokenizer tokens = new StringTokenizer(fecha, "/");
```

```
this.dia = Integer.parseInt(tokens.nextToken());  
this.mes = Integer.parseInt(tokens.nextToken());  
this.anio = Integer.parseInt(tokens.nextToken());  
  
    return true;  
} else {  
    return false;  
}  
  
}  
  
public static Fecha obtenerMayorFecha(Fecha fecha1, Fecha fecha2) {  
  
    Fecha mayor = new Fecha();  
  
    if (fecha1.anio > fecha2.anio) {  
        mayor = fecha1;  
    } else if (fecha1.anio < fecha2.anio) {  
        mayor = fecha2;  
    } else if (fecha1.anio == fecha2.anio) {  
  
        if (fecha1.mes > fecha2.mes) {  
            mayor = fecha1;  
        } else if (fecha1.mes < fecha2.mes) {  
            mayor = fecha2;  
        } else if (fecha1.mes == fecha2.mes) {
```

```
        if (fecha1.dia > fecha2.dia) {  
            mayor = fecha1;  
        } else if (fecha1.dia < fecha2.dia) {  
            mayor = fecha2;  
        } else if (fecha1.dia == fecha2.dia) {  
            mayor = null;  
        }  
    }  
}
```

```
    return mayor;  
}
```

```
public static Fecha obtenerFechaHoy() {
```

```
    Calendar calendario = new GregorianCalendar();  
    String fecha_hoy = "";
```

```
    fecha_hoy += calendario.get(Calendar.DATE);  
    fecha_hoy += "/" + (calendario.get(Calendar.MONTH) + 1);  
    fecha_hoy += "/" + calendario.get(Calendar.YEAR);
```

```
    Fecha hoy = new Fecha();  
    hoy.establecerFecha(fecha_hoy);  
    return hoy;  
}
```

```
public static int obtenerDiasEntreFechas(Fecha fecha1, Fecha fecha2) {
```

```
Fecha mayor = obtenerMayorFecha(fecha1, fecha2);
Fecha menor;
int dias = 0;

if (mayor == fecha1) {
    menor = fecha2;
} else {
    menor = fecha1;
}

dias += (mayor.dia - menor.dia);
dias += (mayor.mes - menor.mes) * 30;
dias += (mayor.anio - menor.anio) * 365;

return dias;
}
```

```
public static Fecha avanzarFechaUnDia(Fecha fecha) {
```

```
    Fecha cambio_fecha = fecha;
    //31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31
```

```
    if (cambio_fecha.mes == 2) {
```

```
        if (esAnioBisiesto(cambio_fecha.anio)) {
```

```
            if (cambio_fecha.dia == 29) {
                cambio_fecha.dia = 1;
```

```
        cambio_fecha.mes++;
    } else {
        cambio_fecha.dia++;
    }

} else if (cambio_fecha.dia == 28) {
    cambio_fecha.dia = 1;
    cambio_fecha.mes++;
} else {
    cambio_fecha.dia++;
}

} else if (esMes31(cambio_fecha.mes)) {

    if (cambio_fecha.dia == 31) {
        cambio_fecha.dia = 1;

        if(cambio_fecha.mes == 12)
        {
            cambio_fecha.mes = 1;
            cambio_fecha.anio++;
        }
        else
            cambio_fecha.mes++;
    } else {
        cambio_fecha.dia++;
    }

} else {
```



```
        if (cambio_fecha.dia == 30) {  
            cambio_fecha.dia = 1;  
            cambio_fecha.mes++;  
        } else {  
            cambio_fecha.dia++;  
        }  
    }  
  
    return cambio_fecha;  
}  
  
public static boolean esMes31(int mes) {
```

```
    switch (mes) {  
        case 1:  
        case 3:  
        case 5:  
        case 7:  
        case 8:  
        case 10:  
        case 12:  
            return true;  
        default:  
            return false;  
    }  
}
```

```
public static boolean esAnioBisiesto(int anio) {
```

```
return ((anio % 4 == 0 && anio % 100 != 0) || anio % 400 == 0);  
  
}
```

```
public static Fecha retrocederFechaUnDia(Fecha fecha) {
```

```
    Fecha cambio_fecha = fecha;
```

```
    //31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31
```

```
    int mes_anterior;
```

```
    if(cambio_fecha.mes == 1)
```

```
        mes_anterior = 12;
```

```
    else
```

```
        mes_anterior = cambio_fecha.mes-1;
```

```
    if (cambio_fecha.mes == 3) {
```

```
        // en marzo verivicando que el anterior es 28
```

```
        if (esAnioBisiesto(cambio_fecha.anio)) {
```

```
            if (cambio_fecha.dia == 1) {
```

```
                cambio_fecha.dia = 29;
```

```
                cambio_fecha.mes--;
```

```
            } else {
```

```
                cambio_fecha.dia--;
```

```
            }
```

```
        } else if (cambio_fecha.dia == 1) {
```

```
            cambio_fecha.dia = 28;
```

```
            cambio_fecha.mes--;
```

```

    } else {
        cambio_fecha.dia--;
    }

} else if (esMes31(mes_anterior)) {
    // mira si su antecesor tiene 31 dias
    if (cambio_fecha.dia == 1) {
        cambio_fecha.dia = 31;

        if(cambio_fecha.mes == 1)
        {
            cambio_fecha.mes = 12;
            cambio_fecha.anio--;
        }
        else
            cambio_fecha.mes--;
    } else {
        cambio_fecha.dia--;
    }

} else {
    if (cambio_fecha.dia == 1) {
        cambio_fecha.dia = 30;
        cambio_fecha.mes--;
    } else {
        cambio_fecha.dia--;
    }
}

```

```
        return cambio_fecha;
    }

    @Override
    public String toString() {
        return dia + "/" + mes + "/" + anio;
    }
}
```

## Refactoring

### En cada Clase

- 0.- Agregar los 'TODO' pendientes.
- 1.- Borrar código innecesario o documentarlo si puede ser reutilizable. Dont repeat yourself
- 2.- Corregir los warnings o 'suprimirlos'(alt+enter en el warning) si estamos totalmente seguros que no se ejecutaran. Borrar imports innecesarios. Para mensajes de debug o error usar un TAG y Log.e/d
- 3.- Optimizar las asignaciones. Si solo se asigna una vez entonces no crear una variable local a menos que ayude a la lectura del código.
- 4.- Cambiar de nombre a las variables a un lenguaje comun, entendible. En este caso español.
- 5.- Cambiar a privado todo a menos que tenga una importante consideración.
- 6.- Reparar el layout