

LeeMed/src/com.bebi.leemed/CargarConfiguracion.java

```
package com.bebi.leemed;
/*
 * Clase inicial en la cual se carga la configuracion del usuario leyendo lo
 que se encuentra almacenado en el SharedPreferences
 */
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.preference.PreferenceManager;

public class CargarConfiguracion extends Activity {
    @Override
    protected void onStart() {
        super.onStart();
        SharedPreferences settings = PreferenceManager
            .getDefaultSharedPreferences(this);
        Static.settings = settings.getBoolean("config", false);
        if (Static.settings == false) {
            Intent i = new Intent(CargarConfiguracion.this,
MainActivity.class);
            finish();
            startActivity(i);
        }
        else if (Static.settings == true) {
            Intent i = new Intent(CargarConfiguracion.this,
NoCiegos.class);
            finish();
            startActivity(i);
        }
    }
}
```

LeeMed/src/com.bebi.leemed/Consulta.java

```
package com.bebi.leemed;

import java.io.IOException;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.KeyEvent;
import android.widget.TextView;

public class Consulta extends Activity {

    TextView tv1;
    helper db;
    Contact medicamento;
    String nombreMedicamento = null;
    String barras = null;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_prospecto);
        tv1 = (TextView) findViewById(R.id.textView1);

        ////////////////////////////////////BBDD////////////////////////////////////
        ////////////////////////////////////

        db = new helper(this);
        try {
            db.createDataBase();
        } catch (IOException ioe) {
            throw new Error("Unable to create database");
        }

        db.open();

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

        Bundle extras = getIntent().getExtras();
        nombreMedicamento = extras.getString("nombreMedicamento");
        String campoSeleccionado = (String) extras
            .getCharSequence("campoSeleccionado");
        barras = (String) extras.getString("barras");

        if (nombreMedicamento != null && barras == null) {
            medicamento = db.getContactName(nombreMedicamento);

            if (medicamento != null) {
                if (campoSeleccionado.equals("Qué es?")) {
                    lecturaProspecto(1);
                }
            }
        }
    }
}
```

```

        } else if (campoSeleccionado.equals("Antes de
tomar...")) {
            lecturaProspecto(2);
        } else if (campoSeleccionado.equals("Cómo tomar"))
{
            lecturaProspecto(3);
        } else if (campoSeleccionado.equals("Posibles
efectos adversos")) {
            lecturaProspecto(4);
        } else if
(campoSeleccionado.equals("Conservación")) {
            lecturaProspecto(5);
        } else if (campoSeleccionado.equals("Información
adicional")) {
            lecturaProspecto(6);
        }
    } else {
        lecturaProspecto(7);
    }
} else if (nombreMedicamento == null && barras != null) {
    medicamento = db.getContactCode(barras);
    if (medicamento != null) {
        tv1.setText(medicamento.getNombre() +
medicamento.getQueEs()
            + medicamento.getAntesDeTomar()
            + medicamento.getComoTomar()
            + medicamento.getEfectosAdversos()
            + medicamento.getConservacion()
            +
medicamento.getInformacionAdicional());
    } else {
        lecturaProspecto(7);
    }
}
}

public void lecturaProspecto(int i) {

    switch (i) {

        case 1:
            tv1.setText(medicamento.getQueEs());
            break;
        case 2:
            tv1.setText(medicamento.getAntesDeTomar());
            break;
        case 3:
            tv1.setText(medicamento.getComoTomar());
            break;
        case 4:
            tv1.setText(medicamento.getEfectosAdversos());
            break;
        case 5:
            tv1.setText(medicamento.getConservacion());
            break;
        case 6:
            tv1.setText(medicamento.getInformacionAdicional());
    }
}

```

```
                break;
            case 7:
                tv1.setText("MEDICAMENTO NO DISPONIBLE");
                break;
        }
    }

    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        switch (keyCode) {
            case KeyEvent.KEYCODE_BACK:

                Intent i = new Intent(Consulta.this, NoCiegos.class);
                finish();
                startActivity(i);

                return true;
        }
        return super.onKeyDown(keyCode, event);
    }
}
```

LeeMed/src/com.bebi.leemed/Contact.java

```
package com.bebi.leemed;

public class Contact {

    // private variables
    int _codigo;
    String _nombre;
    String _que_es;
    String _antes_de_tomar;
    String _como_tomar;
    String _efectos_adversos;
    String _conservacion;
    String _informacion_adicional;
    String _barras;

    public Contact() {}

    // constructor
    public Contact(int codigo, String nombre, String que_es,
efectos_adversos,
        String antes_de_tomar, String como_tomar, String
barras) {
        this._codigo = codigo;
        this._nombre = nombre;
        this._que_es = que_es;
        this._antes_de_tomar = antes_de_tomar;
        this._como_tomar = como_tomar;
        this._efectos_adversos = efectos_adversos;
        this._conservacion = conservacion;
        this._informacion_adicional = informacion_adicional;
        this._barras = barras;
    }

    // constructor
    public Contact(String nombre, String que_es, String antes_de_tomar,
conservacion,
        String como_tomar, String efectos_adversos, String
        String informacion_adicional, String barras) {
        this._nombre = nombre;
        this._que_es = que_es;
        this._antes_de_tomar = antes_de_tomar;
        this._como_tomar = como_tomar;
        this._efectos_adversos = efectos_adversos;
        this._conservacion = conservacion;
        this._informacion_adicional = informacion_adicional;
        this._barras = barras;
    }

    // getting ID
    public int getCodigo() {
        return this._codigo;
    }
}
```

```
// setting id
public void setCodigo(int codigo) {
    this._codigo = codigo;
}

// getting name
public String getNombre() {
    return this._nombre;
}

// setting name
public void setNombre(String nombre) {
    this._nombre = nombre;
}

// getting que_es
public String getQueEs() {
    return this._que_es;
}

// setting que_es
public void setQueEs(String que_es) {
    this._que_es = que_es;
}

// getting antes_de_tomar
public String getAntesDeTomar() {
    return this._antes_de_tomar;
}

// setting antes_de_tomar
public void setAntesDeTomar(String antes_de_tomar) {
    this._antes_de_tomar = antes_de_tomar;
}

// getting ComoTomar
public String getComoTomar() {
    return this._como_tomar;
}

// setting ComoTomar
public void setComoTomar(String como_tomar) {
    this._como_tomar = como_tomar;
}

// getting EfectosAdversos
public String getEfectosAdversos() {
    return this._efectos_adversos;
}

// setting EfectosAdversos
public void setEfectosAdversos(String efectos_adversos) {
    this._efectos_adversos = efectos_adversos;
}

// getting Conservacion
public String getConservacion() {
    return this._conservacion;
}
}
```

```
// setting Conservacion
public void setConservacion(String conservacion) {
    this._conservacion = conservacion;
}

// getting InformacionAdicional
public String getInformacionAdicional() {
    return this._informacion_adicional;
}

// setting InformacionAdicional
public void setInformacionAdicional(String informacion_adicional) {
    this._informacion_adicional = informacion_adicional;
}

// getting barras
public String getbarras() {
    return this._barras;
}

// setting barras
public void setbarras(String barras) {
    this._barras = barras;
}
}
```

LeeMed/src/com.bebi.leemed/helper.java

```
package com.bebi.leemed;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.List;

import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class helper extends SQLiteOpenHelper {

    private static String DB_PATH =
    "/data/data/com.bebi.leemed/databases/";
    private static final String BD_NAME = "BBDDLeeMed";
    private static final String TABLE_MEDICAMENTOS = "medicamentos";
    private static final String KEY_CODIGO = "codigo";
    private static final String KEY_NOMBRE = "nombre";
    private static final String KEY_QUE_ES = "que_es";
    private static final String KEY_ANTES_DE_TOMAR = "antes_de_tomar";
    private static final String KEY_COMO_TOMAR = "como_tomar";
    private static final String KEY_EFECTOS_ADVERSOS = "efectos_adversos";
    private static final String KEY_CONSERVACION = "conservacion";
    private static final String KEY_INFORMACION_ADICIONAL =
    "informacion_adicional";
    private static final String KEY_BARRAS = "barras";
    private SQLiteDatabase myDataBase;
    private final Context myContext;

    public helper(Context contexto) {

        super(contexto, BD_NAME, null, 1);
        this.myContext = contexto;
    }

    //Crea una base de datos vacia en el sistema y la reescribe con
nuestro fichero de base de datos
    public void createDataBase() throws IOException {

        boolean dbExist = checkDataBase();

        if (dbExist) {
            Log.d("Reading: ", "si existe");
            // la base de datos existe y no hacemos nada.
        } else {

```


// Llamando a este metodo se crea la base de datos vacia en la ruta por defecto del sistema
// de nuestra aplicacion por lo que podremos sobreescrirla con nuestra base de datos

```
Log.d("Reading: ", "no si existe");  
this.getReadableDatabase();  
  
try {  
    copyDataBase();  
} catch (IOException e) {  
    throw new Error("Error copiando Base de Datos");  
}  
}
```

//Comprueba si la base de datos existe para evitar copiar siempre el fichero cada vez que se abra la aplicacion.
//@return true si existe, false si no existe

```
private boolean checkDataBase() {
```

```
    SQLiteDatabase checkDB = null;
```

```
    try {
```

```
        String myPath = DB_PATH + BD_NAME;  
        checkDB = SQLiteDatabase.openDatabase(myPath, null,  
            SQLiteDatabase.OPEN_READONLY);
```

```
    } catch (SQLException e) {
```

// si llegamos aqui es porque la base de datos no existe todavia.

```
    }  
    if (checkDB != null) {  
        checkDB.close();  
    }  
    return checkDB != null ? true : false;  
}
```

//Copia nuestra base de datos desde la carpeta assets a la recién creada base de datos en la carpeta de sistema, desde donde podremos acceder a ella.

//Esto se hace con bytestream.

```
private void copyDataBase() throws IOException {
```

```
    // Abrimos el fichero de base de datos como entrada  
    InputStream myInput = myContext.getAssets().open(BD_NAME);  
    // Ruta a la base de datos vacía recién creada  
    String outFileName = DB_PATH + BD_NAME;  
    // Abrimos la base de datos vacía como salida  
    OutputStream myOutput = new FileOutputStream(outFileName);  
    // Transferimos los bytes desde el fichero de entrada al de
```

salida

```
    byte[] buffer = new byte[1024];  
    int length;  
    while ((length = myInput.read(buffer)) > 0) {
```

```

        myOutput.write(buffer, 0, length);
    }
    // Liberamos los streams
    myOutput.flush();
    myOutput.close();
    myInput.close();
}

public void open() throws SQLException {

    // Abre la base de datos
    try {
        createDataBase();
    } catch (IOException e) {
        throw new Error("Ha sido imposible crear la Base de
Datos");
    }
    String myPath = DB_PATH + BD_NAME;
    myDataBase = SQLiteDatabase.openDatabase(myPath, null,
        SQLiteDatabase.OPEN_READONLY);
}

@Override
public synchronized void close() {
    if (myDataBase != null)
        myDataBase.close();
    super.close();
}

@Override
public void onCreate(SQLiteDatabase arg0) {
    // TODO Auto-generated method stub
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    // TODO Auto-generated method stub
}

//Obtenemos un medicamento por codigo de barras
Contact getContactCode(String barras) {
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = db
        .query(TABLE_MEDICAMENTOS, new String[] {
KEY_CODIGO,
KEY_ANTES_DE_TOMAR,
KEY_CONSERVACION,
}, KEY_BARRAS
String.valueOf(barras) }, null,
        KEY_NOMBRE, KEY_QUE_ES,
        KEY_COMO_TOMAR, KEY_EFECTOS_ADVERSOS,
        KEY_INFORMACION_ADICIONAL, KEY_BARRAS
+ "=?", new String[] {
        null, null, null});
    if (cursor.moveToFirst()) {
        cursor.moveToFirst();
    }
}

```

```

        Contact contact = new Contact(
            Integer.parseInt(cursor.getString(0)),
cursor.getString(1),
            cursor.getString(2), cursor.getString(3),
            cursor.getString(4), cursor.getString(5),
            cursor.getString(6), cursor.getString(7),
            cursor.getString(8));
        // return contact
        return contact;
    } else {
        return null;
    }
}

//Obtenemos un medicamento por nombre
Contact getContactName(String name) {
    Log.e("", "Entrando en getContactName");
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = db.query(TABLE_MEDICAMENTOS, new String[] {
KEY_CODIGO,
        KEY_NOMBRE, KEY_QUE_ES, KEY_ANTES_DE_TOMAR,
KEY_COMO_TOMAR,
        KEY_EFECTOS_ADVERSOS, KEY_CONSERVACION,
KEY_INFORMACION_ADICIONAL, KEY_BARRAS }, KEY_NOMBRE
+ "=?",
        new String[] { String.valueOf(name) }, null, null,
null, null);
    if (cursor.moveToFirst()) {
        // cursor.moveToFirst();
        Contact contact = new Contact(
            Integer.parseInt(cursor.getString(0)),
cursor.getString(1),
            cursor.getString(2), cursor.getString(3),
            cursor.getString(4), cursor.getString(5),
            cursor.getString(6), cursor.getString(7),
            cursor.getString(8));
        // return contact
        return contact;
    } else {
        return null;
    }
}

}

// Getting All Contacts
public List<Contact> getAllContacts() {
    List<Contact> contactList = new ArrayList<Contact>();
    // Select All Query
    String selectQuery = "SELECT * FROM medicamentos";
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);

    // looping through all rows and adding to list
    if (cursor.moveToFirst()) {
        do {
            Contact contact = new Contact();

            contact.setCodigo(Integer.parseInt(cursor.getString(0)));

```

```

        contact.setNombre(cursor.getString(1));
        contact.setQueEs(cursor.getString(2));
        contact.setAntesDeTomar(cursor.getString(3));
        contact.setComoTomar(cursor.getString(4));
        contact.setEfectosAdversos(cursor.getString(5));
        contact.setConservacion(cursor.getString(6));

contact.setInformacionAdicional(cursor.getString(7));
        // Adding contact to list
        contactList.add(contact);
    } while (cursor.moveToNext());
    }
    // return contact list
    return contactList;
}

// Getting contacts Count
public int getContactsCount() {
    String countQuery = "SELECT * FROM " + TABLE_MEDICAMENTOS;
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(countQuery, null);
    cursor.close();
    // return count
    return cursor.getCount();
}
}

```

LeeMed/src/com.bebi.leemed/Lector.java

```
package com.bebi.leemed;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

public class Lector extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Creamos el intent con la accion SCAN que se encontrara en el
        interior
        // de CaptureActivity
        Intent intent = new
Intent("com.google.zxing.client.android.SCAN");
        // Le pasamos como extra los modos "SCAN_MODE", "QR_CODE_MODE"
        intent.putExtra("SCAN_MODE", "PRODUCT_MODE");
        // star de CaptureActivity para que nos de resultados en
        // (onActivityResult)
        startActivityForResult(intent, 0);
    }

    // Obtenemos los resultados recibido de
    "startActivityForResult(intent, 0)"
    public void onActivityResult(int requestCode, int resultCode, Intent
intent) {
        if (requestCode == 0) {
            if (resultCode == RESULT_OK) {
                String contents =
intent.getStringExtra("SCAN_RESULT");
                String format =
intent.getStringExtra("SCAN_RESULT_FORMAT");
                Intent i = new Intent(Lector.this, Consulta.class);
                i.putExtra("barras", contents);
                finish();
                startActivity(i);
            } else if (resultCode == RESULT_CANCELED) {
                // Si no obtiene resultados se finaliza asi misma y
                luego arranca el Main (Speech)
                if (Static.settings == false) {
                    Intent i = new Intent(Lector.this,
MainActivity.class);
                    finish();
                    startActivity(i);
                }
            } else if (Static.settings == true) {
                Intent i = new Intent(Lector.this,
NoCiegos.class);
                finish();
                startActivity(i);
            }
        }
    }
}
```

}
}
}
}
}

LeeMed/src/com.bebi.leemed/MainActivity.java

```
package com.bebi.leemed;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

import android.app.Activity;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.content.pm.ResolveInfo;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.speech.tts.TextToSpeech;
import android.util.Log;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageButton;
import android.widget.Toast;

public class MainActivity extends Activity implements
    TextToSpeech.OnInitListener {
    private SpeechRecognizer speechRecognizer;
    private final int SPEECHTOTEXT = 1;
    private final int TEXTTOSPEECH = 2;
    private final int OPCIONES = 3;
    private final int COMIENZO = 0;
    public static final int ERROR\_SPEECH\_TIMEOUT = 6;
    private static final int VOICE\_RECOGNITION\_REQUEST\_CODE = 1;

    TextToSpeech tts;
    helper db;
    Contact medicamento;
    String valorABuscar;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity\_main);

        //
        ////////////////////////////////////////BBDD//////////////////////////////////////
        ////////////////////////////////////////

        db = new helper(this);
        try {
            db.createDataBase();
        } catch (IOException ioe) {
```

```

        throw new Error("Unable to create database");
    }

    db.open();

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

    Intent servicio = new Intent(this, Servicio.class);
    stopService(servicio);

    // Instancia del SpeechRecognizer
    speechRecognizer = SpeechRecognizer
        .createSpeechRecognizer(getBaseContext());

    // Instancia del boton.
    ImageButton btnSpeak = (ImageButton)
    findViewById(R.id.imageButton1);

    // Definiendo la funcionalidad del click del boton.
    OnClickListener onClickListener = new OnClickListener() {
        @Override
        public void onClick(View v) {
            speech(COMIENZO);
        }
    };

    // Setting a click event handler for the button
    btnSpeak.setOnClickListener(onClickListener);
    tts = new TextToSpeech(this, this);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        // lectura del medicamento
        case COMIENZO:
            if (resultCode == RESULT_OK && null != data) {

                String valor = devuelve_string(data);
                if (valor.equals("continuar")) {
                    tts.speak("¿Qué quieres buscar?",
TextToSpeech.QUEUE_FLUSH,
                                null);
                    while (tts.isSpeaking())
                        ;
                    speech(SPEECHTOTEXT);
                } else {
                    if (valor.equals("finalizar")) {
                        finish();
                    }
                    if (!valor.equals("finalizar")
                        && !valor.equals("continuar")) {
                        tts.speak("No te he entendido.
Volvamos a empezar",

```



```

                                TextToSpeech.QUEUE_FLUSH,
null);
                                Intent intent = new Intent(this,
MainActivity.class);
                                startActivity(intent);
                                leer();
                                }
                                }
                                }
                                break;
case SPEECHTOTEXT:
    if (resultCode == RESULT_OK && null != data) {
        String valor = devuelve_string(data);
        valorABuscar = valor;

        if (valor.equals("finalizar")) {
            finish();
        }
        if (!valor.equals("finalizar")) {
            tts.speak("Quisiste decir: " + valor,
                TextToSpeech.QUEUE_FLUSH, null);
            while (tts.isSpeaking())
                ;
            speech(TEXTTOSPEECH);
        }
    }
    break;
// confirmacion del medicamento
case TEXTTOSPEECH:
    if (resultCode == RESULT_OK && null != data) {
        String valor = devuelve_string(data);
        // Toast.makeText(this, "Esto es lo que contiene
valor: " +
        // valor, Toast.LENGTH_SHORT).show();
        //Log.i("Consulta.class", "Esto es lo que contiene
valor:"
        // + valor);
        if (valor.equals("no") || valor.equals("No")) {
            tts.speak("Volvamos a intentarlo. ¿Qué
quieres buscar?",
                TextToSpeech.QUEUE_FLUSH, null);
            while (tts.isSpeaking())
                ;
            speech(SPEECHTOTEXT);
        }
        if (valor.equals("si") || valor.equals("sí")
            || valor.equals("Si") ||
valor.equals("Sí")) {
            // Nos traemos el contacto en funcion del
nombre.

            medicamento =
db.getContactName(valorABuscar);
            if (medicamento == null) {
                tts.speak(
                    "Este medicamento no se
encuentra disponible. Volvamos a intentarlo. ¿Qué quieres buscar?",
                    TextToSpeech.QUEUE_FLUSH,
null);

```



```

        if (valor.equals("1")) {
            lecturaProspecto(1);
        } else if (valor.equals("2")) {
            lecturaProspecto(2);
        } else if (valor.equals("3")) {
            lecturaProspecto(3);
        } else if (valor.equals("4")) {
            lecturaProspecto(4);
        } else if (valor.equals("5")) {
            lecturaProspecto(5);
        } else if (valor.equals("6")) {
            lecturaProspecto(6);
        } else if (valor.equals("finalizar"))
            finish();
        else if (!valor.equals("1") || !valor.equals("2")
            || !valor.equals("3") ||
!valor.equals("4")
            || !valor.equals("5") ||
!valor.equals("6")
            || !valor.equals("finalizar")) {
            tts.speak("No te he entendido. ¿Me lo puedes
repetir?",
                    TextToSpeech.QUEUE_FLUSH, null);
            while (tts.isSpeaking())
                ;
            speech(OPCIONES);
        }
    }
}
break;
}
}

// devuelve la string q recoge el reconocimiento de voz
public String devuelve_string(Intent dato) {
    ArrayList<String> text = dato

.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
    //Log.e("", "ESTO ES LO QUE TIENE LA VARIABLE TEXT" + text);
    String valor = text.get(0);
    //Log.e("", "ESTO ES LO QUE TIENE LA VARIABLE VALOR" + valor);
    return valor;
}

public void onInit(int status) {
    leer();
}

// lectura inicial de la app
public void leer() {
    Locale loc = new Locale("es", "", "");
    if (tts.isLanguageAvailable(loc) >= TextToSpeech.LANG_AVAILABLE)
{
        tts.setLanguage(loc);
    }
    tts.speak(
        "Bienvenido a LEEMED. Si quieres empezar a buscar
di, continuar. Recuerda que en cualquier momento puedes salir diciendo,
finalizar. Pulsa la pantalla para hablar",
        TextToSpeech.QUEUE_FLUSH, null);
}
}

```

```

    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Intent servicio = new Intent(this, Servicio.class);
        startService(servicio);
        tts.shutdown();
    }

    @Override
    public void onResume() {
        super.onDestroy();
    }

    // genera el reconocimiento de voz
    public void speech(int i) {
        Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

        // Getting an instance of PackageManager
        PackageManager pm = getPackageManager();

        // Querying Package Manager
        List<ResolveInfo> activities = pm.queryIntentActivities(intent,
0);

        if (activities.size() <= 0) {
            Toast.makeText(
                getBaseContext(),
                "No Activity found to handle the action
ACTION_RECOGNIZE_SPEECH",
                Toast.LENGTH_SHORT).show();
            return;
        }

        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
            "Voice recognition Demo...");
        startActivityForResult(intent, i);
    }

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

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.item1:
                Intent i = new Intent(this, MenuConfiguracion.class);
                startActivity(i);
            case R.id.item2:
                finish();
        }
    }

```

```

    }
    return true;
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    switch (keyCode) {
        case KeyEvent.KEYCODE_BACK:

            finish();

            return true;
    }
    return super.onKeyDown(keyCode, event);
}

public void prueba() {

    medicamento = db.getContactName(valorABuscar);
}

public void lecturaProspecto(int i) {

    String resultado;
    switch (i) {

        case 1:
            resultado = medicamento.getQueEs();
            tts.speak(resultado, TextToSpeech.QUEUE_FLUSH, null);
            break;
        case 2:
            resultado = medicamento.getAntesDeTomar();
            tts.speak(resultado, TextToSpeech.QUEUE_FLUSH, null);
            break;
        case 3:
            resultado = medicamento.getComoTomar();
            tts.speak(resultado, TextToSpeech.QUEUE_FLUSH, null);
            break;
        case 4:
            resultado = medicamento.getEfectosAdversos();
            tts.speak(resultado, TextToSpeech.QUEUE_FLUSH, null);
            break;
        case 5:
            resultado = medicamento.getConservacion();
            tts.speak(resultado, TextToSpeech.QUEUE_FLUSH, null);
            break;
        case 6:
            resultado = medicamento.getInformacionAdicional();
            tts.speak(resultado, TextToSpeech.QUEUE_FLUSH, null);
            break;
    }
}
}
}

```

LeeMed/src/com.bebi.leemed/MenuConfiguracion.java

```
package com.bebi.leemed;

/*
 * En este .class se lleva a cabo la implementacion de:
 * 1. La vista y funcionalidad del dialogo de alerta que permite cambiar de
version Ciegos/NoCiegos
 * 2. Se guarda la configuracion del usuario mediante el uso de la clase
SharedPreferences
 */

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.view.KeyEvent;
import android.view.View;
import android.widget.RadioButton;

public class MenuConfiguracion extends Activity {

    private RadioButton radiobutton1, radiobutton2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_configuracion);
        radiobutton1 = (RadioButton) findViewById(R.id.radioButton1);
        radiobutton1.setOnClickListener(new
        ButtonClickHandler_Ciegos());
        radiobutton2 = (RadioButton) findViewById(R.id.radioButton2);
        radiobutton2.setOnClickListener(new
        ButtonClickHandler_NoCiegos());
    }

    // Dialogo de alerta en el cual se acepta/cancela cambiar de version
    // (estando en Ciegos)
    public class ButtonClickHandler_Ciegos implements View.OnClickListener
    {
        public void onClick(View view) {
            Dialog vDialog = new
            AlertDialog.Builder(MenuConfiguracion.this)
                .setTitle("¿Quieres cambiar de versión?")
                .setPositiveButton("Aceptar",
                    new
                    DialogButtonClickListener_Ciegos())
                .setNegativeButton("Cancelar",
                    new
                    DialogButtonClickListener_Ciegos()).create();
            vDialog.show();
        }
    }
}
```

```

    }

    // Dialogo de alerta en el cual se acepta/cancela cambiar de version
    // (estando en NoCiegos)
    public class ButtonClickHandler_NoCiegos implements
View.OnClickListener {
        public void onClick(View view) {
            Dialog vDialog = new
AlertDialog.Builder(MenuConfiguracion.this)
                .setTitle("¿Quieres cambiar de versión?")
                .setPositiveButton("Aceptar",
                    new
DialogButtonClickListener_NoCiegos())
                .setNegativeButton("Cancelar",
                    new
DialogButtonClickListener_NoCiegos()).create();
            vDialog.show();
        }
    }

    // Funcionalidad de los botones aceptar/cancelar del dialogo de alerta
    // (estando en Ciegos)
    public class DialogButtonClickListener_Ciegos implements
DialogInterface.OnClickListener {
        public void onClick(DialogInterface dialog, int clicked) {
            switch (clicked) {
                case DialogInterface.BUTTON_POSITIVE:
                    Intent i = new Intent(MenuConfiguracion.this,
                        MainActivity.class);
                    Static.settings = false;
                    guardarConfiguracion();
                    finish();
                    startActivity(i);
                    break;
                case DialogInterface.BUTTON_NEGATIVE:
                    //Aqui iria el codigo en caso de que el boton
cancelar implicase alguna accion
                    break;
            }
        }
    }

    // Funcionalidad de los botones aceptar/cancelar del dialogo de alerta
    // (estando en NoCiegos)
    public class DialogButtonClickListener_NoCiegos implements
DialogInterface.OnClickListener {
        public void onClick(DialogInterface dialog, int clicked) {
            switch (clicked) {
                case DialogInterface.BUTTON_POSITIVE:
                    Intent i = new Intent(MenuConfiguracion.this,
NoCiegos.class);
                    Static.settings = true;
                    guardarConfiguracion();
                    finish();
                    startActivity(i);
                    break;
                case DialogInterface.BUTTON_NEGATIVE:
                    //Aqui iria el codigo en caso de que el boton
cancelar implicase alguna accion

```

```

        break;
    }
}

// Metodo en el cual se implementa un manejador de preferencias
// haciendo uso
// de la clase SharedPreferences
public void guardarConfiguracion() {
    SharedPreferences settings = PreferenceManager

.getDefaulSharedPreferences(MenuConfiguracion.this);
    settings = PreferenceManager

.getDefaulSharedPreferences(MenuConfiguracion.this);
    SharedPreferences.Editor editor = settings.edit();
    editor.putBoolean("config", Static.settings);
    editor.commit();
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    switch (keyCode) {
        case KeyEvent.KEYCODE_BACK:

            if (Static.settings == false) {
                Intent i = new Intent(MenuConfiguracion.this,
                    MainActivity.class);
                finish();
                startActivity(i);
            }
            else if (Static.settings == true) {
                Intent i = new Intent(MenuConfiguracion.this,
NoCiegos.class);

                finish();
                startActivity(i);
            }
            return true;
        }
    }
    return super.onKeyDown(keyCode, event);
}
}

```


LeeMed/src/com.bebi.leemed/NoCiegos.java

```
package com.bebi.leemed;

/*
 * En este .class se lleva a cabo la implementacion de:
 * 1. el cuadro de dialogo que se ejecuta a traves del boton "buscar"
 * 2. el acceso al lector de codigo de barras mediante su boton
correspondiente
 * 3. filtrado del teclado en la entrada del editText
 */

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.text.InputType;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Spinner;

public class NoCiegos extends Activity {

    protected CharSequence[] OpcionesBuscar = { "Qué es?", "Antes de
tomar...",
        "Cómo tomar", "Posibles efectos adversos", "Conservación",
        "Información adicional" }; // Ciegos
    protected ImageButton botonBarras;
    ImageButton botonLector;
    Spinner ListaOpcionesBusqueda; // No ciegos
    EditText et;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_interfaz);
        botonBarras = (ImageButton) findViewById(R.id.imageButton1);
        botonLector = (ImageButton) findViewById(R.id.imageButton2);
        botonBarras.setOnClickListener(new ButtonClickListener());
        botonLector.setOnClickListener(new ClickLector());
        ListaOpcionesBusqueda = (Spinner) findViewById(R.id.spinner1);
        et = (EditText) findViewById(R.id.editText1);

        /* **** FILTRADO DE LA ENTRADA DEL TECLADO **** */
        ListaOpcionesBusqueda
            .setOnItemSelectedListener(new
OnItemSelectedListener() {
                @Override
```

```

        public void onItemClick(AdapterView<?>
arg0, View arg1,
        int arg2, long arg3) {
            String Opcion =
ListaOpcionesBusqueda.getSelectedItemAt()
                .toString();
            //Cuando el Spinner tenia más de una
opción se utilizaba esta parte del código
            //para filtrar los campos de texto de
las diferentes opciones por la que podías buscar
            /*if (Opcion.equals("Código de
barras")) {
et.setInputType(InputType.TYPE_CLASS_NUMBER);
            } else if ((Opcion.equals("Nombre del
medicamento"))) {
et.setInputType(InputType.TYPE_CLASS_TEXT);
            }*/
            if ((Opcion.equals("Nombre del
medicamento"))) {
                et.setInputType(InputType.TYPE_CLASS_TEXT);
            }
            }
            @Override
            public void onNothingSelected(AdapterView<?>
arg0) {
                // TODO Auto-generated method stub
            }
        });
    }
    /*
    * IMPLEMENTACION DEL CLICK DEL BOTON QUE ABRE EL C◊DIGO DE BARRAS EN
LA
    * VERSION PARA NO CIEGOS
    */
    // Funcion del boton "codigo de barras"
    public class ClickLector implements View.OnClickListener {
        public void onClick(View view) {
            pruebaIntent();
        }
    }
    public void pruebaIntent() {
        finish();
        Intent i = new Intent(this, Lector.class);
        startActivity(i);
    }
    /*
    * DIALOGO DE BÚSQUEDA, MOSTRADO A TRAVES DEL BOTON BUSCAR EN LA
    * VERSION PARA NO CIEGOS
    */
    // Genera la estructura del dialogo
    public class ButtonClickHandler implements View.OnClickListener {
        public void onClick(View view) {

```

```

        AlertDialog.Builder builder = new
AlertDialog.Builder(NoCiegos.this);
        builder.setTitle("¿Qué quieres buscar?");
        builder.setSingleChoiceItems(OpcionesBuscar, -1,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface
dialog, int item) {

                    dialog.cancel();
                    if (et.getText().equals("")) {
                        // Posible aviso de campo
                        sin rellenar
                    } else {
                        Intent i = new
Intent(NoCiegos.this,
                Consulta.class);

                i.putExtra("nombreMedicamento", et.getText()
                .toString());

                i.putExtra("campoSeleccionado",
                OpcionesBuscar[item]);

                finish();
                startActivity(i);
            }
        });
        AlertDialog alert = builder.create();
        alert.show();
    }
}

// Metodo autogenerado para que funcione el setMultiChoiceItems();
public class DialogSelectionClickHandler implements
DialogInterface.OnMultiChoiceClickListener {
    public void onClick(DialogInterface dialog, int clicked,
        boolean selected) {
    }
}

// Funcion del boton "buscar"
public class DialogButtonClickHandler implements
DialogInterface.OnClickListener {
    public void onClick(DialogInterface dialog, int clicked) {
        switch (clicked) {
            case DialogInterface.BUTTON_POSITIVE:
                finish();
                break;
        }
    }
}

/* IMPLEMENTACION DEL MENU PRINCIPAL (BOTON DEL TELEFONO) */
// Genera la estructura del menu
@Override
public boolean onCreateOptionsMenu(Menu menu) {

```

```

        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu, menu);
        return true;
    }

    // Funcion de los botones del menu
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.item1:
                Intent i = new Intent(this, MenuConfiguracion.class);
                startActivity(i);

            case R.id.item2:
                finish();
        }
        return true;
    }

    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        switch (keyCode) {
            case KeyEvent.KEYCODE_BACK:
                finish();
                return true;
        }
        return super.onKeyDown(keyCode, event);
    }
}

```

LeeMed/src/com.bebi.leemed/Recibidor.java

```
package com.bebi.leemed;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class Recibidor extends BroadcastReceiver {

    public void onReceive(Context context, Intent intent) {

        Intent servicio = new Intent();
        servicio.setAction("com.bebi.arrancandoconelsistema.Servicio");
        context.startService(servicio);
    }
}
```

LeeMed/src/com.bebi.leemed/RecibidorBloqueado.java

```
package com.bebi.leemed;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class RecibidorBloqueado extends BroadcastReceiver {
    private boolean screenOff;

    public void onReceive(Context context, Intent intent) {

        if (intent.getAction().equals(Intent.ACTION_SCREEN_OFF)) {
            screenOff = true;
        } else if (intent.getAction().equals(Intent.ACTION_SCREEN_ON)) {
            screenOff = false;
        }
        // El recibidor estara pendiente en todo momento de si la luz de
la pantalla esta
        // encendida o apagada, en base a eso iniciara el servicio
pasandole un extra que
        // que sera la variable screenOff con nombre de screen_state.
        Intent i = new Intent(context, Servicio.class);
        i.putExtra("screen_state", screenOff);
        context.startService(i);
    }
}
```

LeeMed/src/com.bebi.leemed/Servicio.java

```
package com.bebi.leemed;

import android.app.Service;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.IBinder;
import android.util.Log;
import android.widget.Toast;

public class Servicio extends Service implements SensorEventListener {
    private int todoBien;

    @Override
    public void onCreate() {
        super.onCreate();
        // Creamos un IntentFilter manualmente con la accion para
        Pantalla Encendida
        IntentFilter filter = new IntentFilter(Intent.ACTION_SCREEN_ON);
        // Le añadimos la accion Pantalla Apagada
        filter.addAction(Intent.ACTION_SCREEN_OFF);
        // Nos creamos un recibidor llamando a nuestro constructor
        BroadcastReceiver mReceiver = new RecibidorBloqueado();
        // Registramos el recibidor pasandole el objeto y el filtro.
        registerReceiver(mReceiver, filter);
        // Se accede a los sensores del sistema
        SensorManager manager = (SensorManager)
        getSystemService(Context.SENSOR_SERVICE);
        // Se accede al acelerometro
        Sensor acelerometro =
        manager.getSensorList(Sensor.TYPE_ACCELEROMETER)
            .get(0);
        // Registro del acelerometro en modo juegos para leer datos en
        tiempo
        // real
        if (!manager.registerListener(this, acelerometro,
            SensorManager.SENSOR_DELAY_GAME)) {
            Toast.makeText(this, "No se ha registrado el sensor",
                Toast.LENGTH_SHORT);
        }
        Toast.makeText(this, "Servicio creado",
            Toast.LENGTH_SHORT).show();
    }

    @Override
    // En este caso solo usamos en parametro intent, pero si quisieramos
    usar
```

```

    // los otros dos, saber que: flags nos da información sobre como
comienza la solicitud. Puede ser 0,
    // START_FLAG_REDELIVERY o START_FLAG_RETRY
    // Un valor distinto de cero se utiliza cuando se reinicia un servicio
tras
    // haber sido destruido.
    // id arranque es un entero unico representando la solicitud de
arranque
    // especifica. es decir si el servicio
    // es la segunda vez que se arranca su id sera 2, si es la tercera
será 3,
    // etc...
    public int onStartCommand(Intent intent, int flags, int startId) {
        // Log.i("Servicio.class", "onStartCommand");
        if (intent == null) {
            // Log.i("Servicio.class", "Intent null");
        } else {
            SensorManager manager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
            boolean screenOn = intent.getBooleanExtra("screen_state",
false);
            if (!screenOn) {
                Sensor acelerometro = manager.getSensorList(
                    Sensor.TYPE_ACCELEROMETER).get(0);
                manager.registerListener(this, acelerometro,
                    SensorManager.SENSOR_DELAY_GAME);
            } else {
                manager.unregisterListener(this);
            }
        }
        // Valor que indica al sistema como tiene que comportarse cuando
el
        // servicio es destruido.
        // En este caso el sistema tratará de crear de nuevo el servicio
cuando
        // disponga de memoria suficiente.
        // Usamos este modo ya que en la llamada a nuestro servicio no se
le pasa
        // ningun dato de importancia.
        // ya que cuando se destruya por ejemplo por falta de memoria y
se
        // vuelva a invocar, este no será invocado
        // con el intent que lo llamo inicialmente sino con un intent
NULL
        // (debido al START_STICKY que le pusimos), como nuestro
servicio no recibe
        // información extra de quien lo invoca no es necesario que
usamos el
        // intent inicial y nos sirve el NULL.
        return START_STICKY;
    }

    @Override
    public void onDestroy() {
        Log.i("Servicio.class", "onDestroy");
        // Una vez el servicio se destruya dejaremos de escuchar el
acelerometro
        // para evitar problemas

```



```

        SensorManager manager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        manager.unregisterListener(this);
        // Una vez el servicio se destruya anularemos el receiver para
evitar
        // problemas
        // BroadcastReceiver mReceiver = new RecibidorBloqueado();
        // unregisterReceiver (mReceiver);

        super.onDestroy();
    }

    @Override
    // Metodo que forma parte de Service (dedicado a comunicaci3n). NO lo
usamos
    // en este caso.
    public IBinder onBind(Intent intent) {
        return null;
    }

    // Metodo registrador de cambios en los sensores.
    // FUNCIONA COMO UN BUCLE!!

    public void onSensorChanged(SensorEvent event) {
        // Reconocimiento del primer punto de interes.
        if (event.values[0] > -4.0 && event.values[0] < 3.0
            && event.values[1] > 6.0 && event.values[1] < 20.0
            && event.values[2] > 4.0 && event.values[2] < 20.0)
{
        event.values[0]); // Log.i("Servicio", "Los valores de x son: " +
        event.values[1]); // Log.i("Servicio", "Los valores de y son: " +
        event.values[2]); // Log.i("Servicio", "Los valores de z son: " +

            todoBien = 1;
        }
        // Reconocimiento del segundo punto de interes.
        if ((event.values[0] > 1.0 && event.values[0] < 7.0)
            && (event.values[1] > -3.0 && event.values[1] <
6.0)
            && event.values[2] < -10.0 && todoBien == 1) {
        event.values[0]); // Log.d("Servicio", "Los valores de x son: " +
        event.values[1]); // Log.d("Servicio", "Los valores de y son: " +
        event.values[2]); // Log.d("Servicio", "Los valores de z son: " +

        startActivity(new Intent(this, MainActivity.class)
            .setFlags(Intent.FLAG_ACTIVITY_NEW_TASK));
        todoBien = 0;
        }
    }
}
/*

```

```

    * Metodo que forma parte de la implementacion de SensorEventListener
y que
    * sirve para cambiar la precision de los sensores (NO lo usamos en
este
    * caso)
    */

    public void onAccuracyChanged(Sensor sensor, int accuracy) {
}
}
```

LeeMed/src/com.bebi.leemed/Static.java

```
package com.bebi.leemed;  
  
public class Static {  
    static boolean settings = false;  
}
```

LeeMed/src/com.bebi.leemed/Widget.java

```
package com.bebi.leemed;

import android.app.PendingIntent;
import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.Context;
import android.content.Intent;
import android.widget.RemoteViews;

public class Widget extends AppWidgetProvider {

    public void onUpdate(Context context, AppWidgetManager
appWidgetManager,
        int[] appWidgetIds) {

        RemoteViews views = new RemoteViews(context.getPackageName(),
            R.layout.activity_widget);
        Intent intent = new Intent(context, MainActivity.class);
        PendingIntent pi = PendingIntent.getActivity(context, 0, intent,
0);

        views.setOnClickPendingIntent(R.id.imageButton1, pi);
        appWidgetManager.updateAppWidget(appWidgetIds[0], views);
    }
}
```

LeeMed/res/layout/activity_configuration.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    android:orientation="vertical" >

    <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/radioButton1"
        android:layout_below="@+id/radioButton1"
        android:layout_marginTop="10dp"
        android:text="@string/radioButton2"
        android:textColor="@color/black" />

    <TextView
        android:id="@+id/TextView02"
        android:layout_width="wrap_content"
        android:layout_height="20sp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:background="@color/naranja"
        android:text="@string/TextView02"
        android:textColor="@color/white"
        android:textSize="15sp"
        android:textStyle="bold"
        android:typeface="sans" />

    <RadioButton
        android:id="@+id/radioButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/TextView02"
        android:layout_marginLeft="18dp"
        android:layout_marginTop="10dp"
        android:text="@string/radioButton1"
        android:textColor="@color/black" />

</RelativeLayout>
```

LeeMed/res/layout/activity_lector.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_gravity="center"
    android:orientation="vertical" >

    <Button
        android:id="@+id/scanner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:gravity="center"
        android:text="Leer código QR" >
    </Button>

    <Button
        android:id="@+id/scanner2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:gravity="center"
        android:text="Leer código de barras" >
    </Button>

</LinearLayout>
```

LeeMed/res/layout/activity_main.xml

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

    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:background="@color/white"
        android:height="80dp"
        android:src="@drawable/microfono"
        android:width="180dp" />

</RelativeLayout>
```

LeeMed/res/layout/activity_nociegos.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:background="@drawable/medicamentos"
    tools:context=".MainActivity" >

    <ImageButton
        android:id="@+id/imageButton2"
        android:layout_width="180sp"
        android:layout_height="80sp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="22dp"
        android:background="@color/white"
        android:src="@drawable/barras" />

    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@+id/editText1"
        android:layout_alignParentLeft="true"
        android:entries="@array/Opciones"
        android:prompt="@string/textoOpciones" />

    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="50sp"
        android:layout_height="50sp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="124dp"
        android:background="@color/white"
        android:src="@drawable/search2" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/imageButton1"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_marginBottom="61dp"
        android:ems="10"
        android:text="@string/editText1"
        android:textColor="@color/naranja"
        android:textStyle="bold"
        android:typeface="sans" />

</RelativeLayout>
```


LeeMed/res/layout/activity_prospecto.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FE9A2E"
    android:orientation="vertical"
    android:padding="15dp" >

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:padding="5dp"
        android:scrollbars="vertical" >

        <TextView
            android:id="@+id/textView1"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:background="@drawable/back"
            android:padding="5dp" />
    </ScrollView>
</LinearLayout>
```

LeeMed/res/layout/activity_widget.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" >

    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:background="@color/white"
        android:height="80dp"
        android:src="@drawable/microfono"
        android:width="180dp" />

</RelativeLayout>
```

LeeMed/res/menu/activity_main.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/menu_settings"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="@string/menu_settings"/>
</menu>
```

LeeMed/res/menu/menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
  <item android:id="@+id/item1" android:title="Configuración"></item>
  <item android:id="@+id/item2" android:title="Salir"></item>

</menu>
```

LeeMed/res/values/colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="white">#FFFFFF</color>
    <color name="yellow">#FFFF00</color>
    <color name="fuchsia">#FF00FF</color>
    <color name="red">#FF0000</color>
    <color name="silver">#C0C0C0</color>
    <color name="gray">#808080</color>
    <color name="olive">#808000</color>
    <color name="purple">#800080</color>
    <color name="maroon">#800000</color>
    <color name="aqua">#00FFFF</color>
    <color name="lime">#00FF00</color>
    <color name="teal">#008080</color>
    <color name="green">#008000</color>
    <color name="blue">#0000FF</color>
    <color name="navy">#000080</color>
    <color name="black">#000000</color>
    <color name="azul_claro">#81F7D8</color>
    <color name="azul_settings">#045FB4</color>
    <color name="naranja">#DF7401</color>
    <color name="naranjita">#FAAC58</color>
</resources>
```

LeeMed/res/values/strings.xml

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

    <string name="app_name">LeeMed</string>
    <string name="menu_settings">Settings</string>
    <string name="editText1"></string>
    <string name="textView1">Nombre del medicamento:</string>
    <string name="radioButton2">Versión para usuarios sin discapacidad
visual</string>
    <string name="TextView02">Tipo de vista</string>
    <string name="radioButton1">Versión para usuarios con discapacidad
visual</string>
    <string name="textoOpciones">Opciones</string>
    <string-array name="Opciones">
        <item >Nombre del medicamento</item>
    </string-array>

    <!-- Strings testeo -->
    <string name="hello_world">Hello world!</string>
    <string name="tv1">ENHORABUENA ARRANQUE!!!</string>
    <string name="textView4">Tipo de apertura</string>
    <string name="checkBox10">Gestual</string>

</resources>
```

LeeMed/res/xml/appwidgetproviderinfo.xml

```
<appwidget-provider
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:minWidth="146dip"
  android:minHeight="294dip"
  android:updatePeriodMillis="0"
  android:resizeMode="horizontal|vertical"
  android:initialLayout="@Layout/activity_widget" />
```

LeeMed/AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.bebi.Leemed"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="16" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/Logo_Leemed"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.bebi.Leemed.CargarConfiguracion"
            android:label="@string/app_name"

            android:screenOrientation="portrait" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name=".Recibidor">
            <intent-filter>
                <action android:name="android.intent.action.BOOT_COMPLETED"/>
            </intent-filter>
        </receiver>
        <receiver android:name=".Widget">
            <intent-filter>
                <action
                    android:name="android.appwidget.action.APPWIDGET_UPDATE"/>
            </intent-filter>
            <meta-data
                android:name="android.appwidget.provider"
                android:resource="@xml/appwidgetproviderinfo"/>
        </receiver>

        <service android:name=".Servicio">
            <intent-filter>
                <action
                    android:name="com.bebi.arrancandoconeelsistema.Servicio"/>
            </intent-filter>
        </service>
        <activity android:name="ArranqueActivity"></activity>
        <activity android:name="Consulta"></activity>
        <activity android:name="Lector"></activity>
        <activity
            android:name="com.bebi.Leemed.MenuConfiguracion"
            android:screenOrientation="portrait">
        </activity>
```



```

<activity
    android:name="MainActivity"
    android:launchMode="singleTop"
    android:screenOrientation="portrait" >
</activity>
<activity
    android:name="NoCiegos"
    android:screenOrientation="portrait">
</activity>
<activity
    android:clearTaskOnLaunch="true"
    android:stateNotNeeded="true"
    android:configChanges="orientation|keyboardHidden"
    android:name="com.google.zxing.client.android.CaptureActivity"
    android:screenOrientation="Landscape"
    android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
    android:windowSoftInputMode="stateAlwaysHidden" android:label="">
    <intent-filter >
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
    <intent-filter >
        <action android:name="com.google.zxing.client.android.SCAN"

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
</application>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.CAMERA"/>
</manifest>

```