

MEMORIA DESCRIPTIVA

“MONITOR DE DOLOR”

Contenido

1. RESUM DE L'APLICACIÓ.....	3
2. LENGUATGE DE PROGRAMACIÓ.....	4
3. ENTORN OPERATIU	4
4. LLISTAT DE FITXERS	5
5. DIAGRAMA DE FLUX.....	6
6. EXECUTABLE DEL PROGRAMA.....	7
7. MANUAL D'USUARI	7
8. TOTALITAT DEL CODI FONT	7

1. RESUM DE L'APLICACIÓ

Esta aplicación para móvil está destinada a investigar cómo influye el dolor en la vida diaria de las personas que sufren dolor crónico. DURANTE 30 DÍAS (una vez por la mañana y otra por la tarde) registra unas breves preguntas relacionadas con el dolor (dolor, cansancio, estado de ánimo, etc).

Además permite registrar adicionalmente cómo se sienten los pacientes de dolor crónico cuando tienen un episodio de dolor agudo. Cuando esto sucede, el paciente puede entrar en la aplicación y responder a las preguntas para episodios de dolor agudo.

Junto con las respuestas de las preguntas, se almacenan también los datos de las variables contextuales (llamadas realizadas, llamadas recibidas, llamadas perdidas, sms enviados y sms recibidos desde que se respondió el último grupo de preguntas) y la información meteorológica del lugar en la que se encuentra el usuario (ciudad, temperatura, presión, humedad, viento y descripción meteorológica en el momento que responde las preguntas).

Para poder recoger la información meteorológica es necesario que el usuario tenga activo el GPS o que permita que las aplicaciones puedan acceder a su localización. Una vez la aplicación a obtenido las coordenadas geográficas del usuario consulta un servidor metereológico para obtener los datos meteorológicos más relevantes.

La APP está formada por cuatro grupos de preguntas:

El primer grupo de preguntas se administra una sola vez en la primera ocasión que el usuario accede a la APP e incluye características demográficas (fecha de nacimiento, género, estado civil, situación laboral y nivel de estudios), información sobre el dolor (tipo de dolor, localización del dolor, persona que hizo el diagnóstico, duración del dolor desde su aparición y tratamiento actual para el dolor) y comorbilidad de depresión y ansiedad.

El segundo grupo de preguntas incluye las preguntas que se administran cada mañana. Estas incluyen una medida de intensidad del dolor, fatiga, estado de ánimo (felicidad, tristeza, ansiedad y rabia), percepción de salud general, interferencia del dolor en el sueño, miedo / evitación del trabajo, catastrofismo (escala de desesperanza), disposición a la actividad con dolor, miedo / evitación de actividades físicas y aceptación.

El tercer grupo de preguntas se administra cada tarde. Este grupo incluye preguntas sobre intensidad del dolor, fatiga, estado de ánimo (felicidad, tristeza, ansiedad y rabia), interferencia del dolor en el trabajo fuera del hogar o el trabajo doméstico, interferencia del dolor en las actividades de ocio, interferencia del dolor en las relaciones con los demás, estrategias de afrontamiento utilizadas ante el dolor, catastrofismo (escala de magnificación), catastrofismo (escala de rumiación), grado de actividad, área de la vida de foco de actividad, uso de medicación de rescate y efectos adversos experimentados.

Finalmente, el cuarto grupo de preguntas se utiliza en la evaluación del dolor agudo, por lo que aparece cada vez que el usuario accede a la aplicación fuera de las horas programadas de evaluación de este modo podremos obtener información en tiempo real de lo que sucede durante los episodios de dolor agudo, permitiéndonos avanzar en el conocimiento que tenemos acerca del dolor crónico. Específicamente, esto nos permitirá

saber la frecuencia de los episodios de dolor agudo que presentan estos pacientes, así como su intensidad y sus consecuencias emocionales. Estas preguntas incluyen medidas de intensidad del dolor, fatiga y estado de ánimo (felicidad, tristeza, ansiedad y rabia).

La selección del contenido de esta APP ha sido realizada siguiendo las recomendaciones de la Iniciativa sobre Métodos, Medidas, y Evaluación del Dolor en Ensayos Clínicos (Initiative on Methods, Measurement, and Pain Assessment in Clinical Trials, IMMPACT) y siguiendo un abordaje biopsicosocial del dolor.

La aplicación se encuentra disponible para la plataforma Android para móvil y tablet, y puede ser descargada de forma gratuita en GooglePlay:

<https://play.google.com/store/apps/details?id=appcontextualv3.code>

Antes de descargarla, aparece una descripción de la aplicación. Esta incluye el nombre y la información de contacto de la Dra García Palacios, el objetivo de la aplicación, el diseño del estudio, cómo se seleccionó el contenido de la aplicación y una declaración garantizando el anonimato de los datos proporcionados. La aplicación ha sido desarrollada siguiendo todas las premisas que establece la LOPD (Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal) para proteger la identidad y el anonimato de todas las personas que la usen.

2. LENGUATGE DE PROGRAMACIÓ

Entorno de desarrollo Eclipse y el lenguaje de programación JAVA.

3. ENTORN OPERATIU

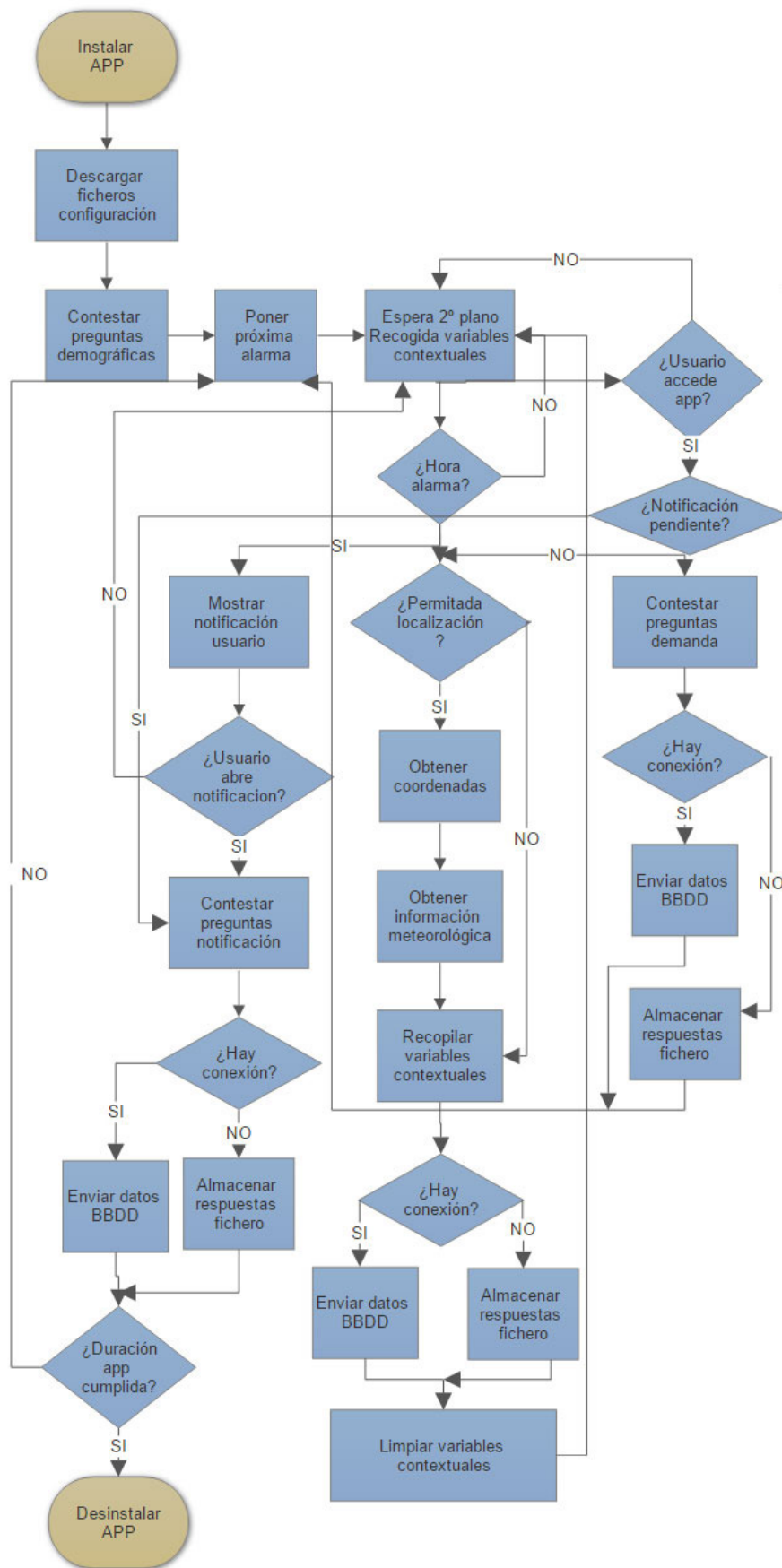
La aplicación "Monitor de Dolor" se ha desarrollado para sistema operativo Android (versión 2.2 en adelante).

4. LLISTAT DE FITXERS

AppContextualv3

- AndroidManifest.xml
- src/appcontextualv3/code
 - CambioHora.java
 - Conf_inicial.java
 - Configuracion.java
 - Demanda.java
 - FinPreguntas.java
 - Instrucciones.java
 - Lanzamiento.java
 - Llamadas.java
 - MyLocationListener.java
 - ObtenerInfoTiempo.java
 - Pregunta.java
 - Receiver_BootCompleted.java
 - RecibeAlarma.java
 - RecibeAlarma2.java
 - Respuesta.java
 - SMS.java
 - Tiempo.java
 - Utils.java
 - Variables.java
 - VisorPregunta.java
 - VisorPreguntaInicial.java
- res/layout
 - activity_cambio_hora.xml
 - conf_inicial.xml
 - fondo.xml
 - instrucciones.xml
 - splash.xml
- res/menú
 - horas1.xml
 - horas2.xml
 - instrucciones1.xml
 - instrucciones2.xml
 - preguntas.xml

5. DIAGRAMA DE FLUX



6. EXECUTABLE DEL PROGRAMA

AppDolorv3.apk

(Se entrega en CD junto con esta memoria.)

7. MANUAL D'USUARI

No existe manual de usuario.

8. TOTALITAT DEL CODI FONT

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="appcontextualv3.code"
    android:versionCode="3"
    android:versionName="3.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/logo2"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="appcontextualv3.code.Demanda"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        <receiver android:name="appcontextualv3.code.Receiver_BootCompleted" >
            <intent-filter>
                <action android:name="android.intent.action.BOOT_COMPLETED" />
            </intent-filter>
        </receiver>
        <receiver android:name="appcontextualv3.code.Llamadas" >
            <intent-filter>
                <action android:name="android.intent.action.PHONE_STATE" />
                <action android:name="android.intent.action.NEW_OUTGOING_CALL" />
            </intent-filter>
        </receiver>
        <receiver android:name="appcontextualv3.code.RecibeAlarma" >
        </receiver>
        <receiver android:name="appcontextualv3.code.RecibeAlarma2" >
        </receiver>

        <service android:name="appcontextualv3.code.SMS" >
            <intent-filter>
                <action android:name="appcontextualv3.code.SMS" />
            </intent-filter>
        </service>

        <activity
            android:name="appcontextualv3.code.VisorPregunta"
            android:screenOrientation="portrait"
            android:theme="@style/Theme.AppCompat.Light.DarkActionBar" >
        </activity>
        <activity
            android:name="appcontextualv3.code.FinPreguntas"
            android:screenOrientation="portrait" >
        </activity>
        <activity
            android:name="appcontextualv3.code.Conf_inicial"
            android:screenOrientation="portrait" >
        </activity>
        <activity
            android:name="appcontextualv3.code.CambioHora"
            android:screenOrientation="portrait"
            android:theme="@style/Theme.AppCompat.Light.DarkActionBar" >
        </activity>
        <activity
            android:name="appcontextualv3.code.Instrucciones"
            android:screenOrientation="portrait"
            android:theme="@style/Theme.AppCompat.Light.DarkActionBar" >
        </activity>
        <activity
```

```
        android:name="appcontextualv3.code.VisorPreguntaInicial"
        android:screenOrientation="portrait" >
    </activity>
</application>

<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_LOGS" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_SUPERUSER" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />

</manifest>
```

```
package appcontextualv3.code;

import java.lang.reflect.Field;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.net.ParseException;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.util.TypedValue;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.WindowManager;
import android.view.ViewGroup.LayoutParams;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.Spinner;
import android.widget.TextView;

public class CambioHora extends AppCompatActivity {

    int origen;
    int numerolanzamientos;
    int duracion=0;
    ProgressBar pb;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.Lay
outParams.FLAG_FULLSCREEN);

        ActionBar actionBar = getSupportActionBar();
        actionBar.setDisplayShowTitleEnabled(false);
        //actionBar.setDisplayShowHomeEnabled(false);
        actionBar.setBackgroundDrawable(new ColorDrawable(Color.parseColor("#333333")));

        origen=getIntent().getExtras().getInt("origen",0);
        setContentView(R.layout.activity_cambio_hora);

        LinearLayout fondo=(LinearLayout)findViewById(R.id.layouthora);
        pb = (ProgressBar) findViewById(R.id.progressBar1);
        pb.setVisibility(View.GONE);

        numerolanzamientos= Receiver_BootCompleted.config.lanzamientos.size();

        List <String> listahoras=new ArrayList<String>();

        listahoras.add("- 2 horas");
        listahoras.add("- 1 hora");
        listahoras.add("No cambiar");
        listahoras.add("+ 1 hora");
```

```
listahoras.add("+ 2 horas");

final Spinner[] ddls= new Spinner[numerolanzamientos];
final String[] horasorigen= new String[numerolanzamientos];
final SharedPreferences settings =
getApplicationContext().getSharedPreferences(Utils.appname, 0);

for (int i=0; i<numerolanzamientos; i++)
{
    LinearLayout ll= new LinearLayout(this);
    ll.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
    LayoutParams.WRAP_CONTENT));
    ll.setGravity(Gravity.CENTER_HORIZONTAL);
    ll.setPadding(5, 10, 5, 5);
    ll.setOrientation(LinearLayout.HORIZONTAL );

    //obtener hora inicial
    horasorigen[i]=settings.getString("hora"+i, "");
    TextView tv=new TextView(this);
    tv.setText("A las "+horasorigen[i]+" horas");
    tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,16);
    tv.setTextColor(Color.parseColor("#676767"));
    tv.setPadding(0, 0, 15, 0);
    ll.addView(tv);

    ddls[i] = new Spinner(this);
    ArrayAdapter<String> spinnerArrayAdapter = new ArrayAdapter<String>( this,
    android.R.layout.simple_spinner_item,listahoras);
    spinnerArrayAdapter.setDropDownViewResource(
    android.R.layout.simple_spinner_dropdown_item );
    ddls[i].setMinimumWidth(220);
    ddls[i].setAdapter(spinnerArrayAdapter);

    //marcar opción seleccionada
    int opcionhora=settings.getInt("opcionhora"+i, 2);
    ddls[i].setSelection(opcionhora);

    ll.addView(ddls[i]);
    fondo.addView(ll);
}

//poner para cambiar la duracion
List <String> listadias=new ArrayList<String>();

listadias.add("26 días");
listadias.add("28 días");
listadias.add("30 días");

LinearLayout ll2= new LinearLayout(this);
ll2.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT));
ll2.setGravity(Gravity.CENTER_HORIZONTAL);
ll2.setPadding(5, 10, 5, 5);
ll2.setOrientation(LinearLayout.HORIZONTAL );

TextView tv2=new TextView(this);
tv2.setText("Duración del estudio");
tv2.setTextSize(TypedValue.COMPLEX_UNIT_DIP,16);
tv2.setTextColor(Color.parseColor("#676767"));
tv2.setPadding(0, 0, 15, 0);
ll2.addView(tv2);

final Spinner ddl = new Spinner(this);
ArrayAdapter<String> spinnerArrayAdapter2 = new ArrayAdapter<String>( this,
android.R.layout.simple_spinner_item,listadias);
```

```
spinnerArrayAdapter2.setDropDownViewResource(  
    android.R.layout.simple_spinner_dropdown_item );  
ddl.setMinimumWidth(220);  
ddl.setAdapter(spinnerArrayAdapter2);
```

```
//marcar opción seleccionada  
int opciondia=settings.getInt("opciondias", 2);  
ddl.setSelection(opciondia);
```

```
ll2.addView(ddl);  
fondo.addView(ll2);
```

```
//boton guardar
```

```
Button salir= new Button(this);  
if (origen==1)  
{  
    salir.setText("Finalizar");  
}  
if (origen==2)  
{  
    salir.setText("Guardar");  
}
```

```
salir.setTextSize(TypedValue.COMPLEX_UNIT_DIP,30);  
salir.setBackgroundResource(R.drawable.botonazul);  
salir.setTextColor(Color.WHITE);  
LinearLayout.LayoutParams params2= new  
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT ,  
    LinearLayout.LayoutParams.WRAP_CONTENT);  
params2.setMargins(10,40,10,10);  
salir.setLayoutParams(params2);  
fondo.addView(salir);
```

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

```
    @Override  
    public void onClick(View v) {
```

```
        SharedPreferences.Editor editor = settings.edit();  
        for (int i=0; i<numerolanzamientos; i++)  
        {
```

```
            int  
            seleccionado=ddls[i].getSelectedItemPosition();  
            editor.putInt("opcionhora"+i, seleccionado);  
            SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm:ss");  
            Calendar cal=Calendar.getInstance();
```

```
            try {  
                cal.setTime( dateFormat.parse(horasorigen[i]));  
            } catch (java.text.ParseException e) {  
                // TODO Auto-generated catch block  
                e.printStackTrace();  
            }  
            switch (seleccionado) {
```

```
                case 0:  
                    cal.add(Calendar.HOUR_OF_DAY, -2);  
                    break;
```

```
                case 1:  
                    cal.add(Calendar.HOUR_OF_DAY, -1);  
                    break;
```

```
                case 3:  
                    cal.add(Calendar.HOUR_OF_DAY, 1);  
                    break;
```

```
                case 4:  
                    cal.add(Calendar.HOUR_OF_DAY, 2);  
                    break;
```

```
                default:  
                    break;
```

```
            }  
            Receiver_BootCompleted.config.lanzamientos.get(i).hora =cal.getTime();
```

```
        }
```

```

//guardar la duracion
Calendar cal=Calendar.getInstance();
int seleccionado2=ddl.getSelectedItemPosition();
editor.putInt("opciondias", seleccionado2);

switch (seleccionado2) {
case 0:
    duracion=26;
    break;
case 1:
    duracion=28;
    break;
case 2:
    duracion=30;
    break;
default:
    break;
}
for (int i=0; i<numerolanzamientos; i++)
{
    cal.setTime(Receiver_BootCompleted.config.lanzamientos.get(i).inicio
    );
    cal.add(Calendar.DAY_OF_MONTH , duracion);
    // cal.set(Calendar.HOUR_OF_DAY , 23);
    // cal.set(Calendar.MINUTE , 59);
    // cal.set(Calendar.SECOND ,59);
    Receiver_BootCompleted.config.lanzamientos.get(i).fin =cal.getTime();
}

cal.setTime(Receiver_BootCompleted.config.fechainicio );
cal.add(Calendar.DAY_OF_MONTH , duracion);
// cal.set(Calendar.HOUR_OF_DAY , 23);
// cal.set(Calendar.MINUTE , 59);
// cal.set(Calendar.SECOND ,59);
Receiver_BootCompleted.config.fechafin =cal.getTime();
editor.commit();

new Wait2().execute("");
}
});
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    if (origen==1)
    {
        getMenuInflater().inflate(R.menu.horas1, menu);
    }

    if (origen==2)
    {
        getMenuInflater().inflate(R.menu.horas2, menu);
    }
    return true;
}

```

```

@Override

```

```

public boolean onOptionsItemSelected ( MenuItem item){
    switch(item.getItemId()){
        case R.id.action_back :
            Intent intent=new Intent(CambioHora.this,VisorPregunta.class);
            intent.putExtra("cont",Receiver_BootCompleted.config.contador );
            CambioHora.this.startActivity(intent);
            CambioHora.this.finish();
            return true;
        case R.id.action_exit :
            CambioHora.this.finish();
            return true;
    }
    return super.onOptionsItemSelected( item);
}

private class Wait2 extends AsyncTask<String, Void, Integer> {

    protected void onPreExecute() {
        // pb.setVisibility(View.VISIBLE);
    }

    protected Integer doInBackground(String... urls) {

        Utils.almacenarObservacionesUsuario(Receiver_BootCompleted.config.imei,
        String.valueOf(duracion));
        return null;

    }

    protected void onPostExecute(Integer valor) {

        // pb.setVisibility(View.GONE);
        if (origen==1)
        {
            /* Intent intent=new Intent(CambioHora.this,Instrucciones.class);
            intent.putExtra("origen",1 );
            CambioHora.this.startActivity(intent);*/
            //volver a poner alarma
            Date dt = new Date();

            if ((dt.after(Receiver_BootCompleted.config.fechainicio)) &&
            (dt.before(Receiver_BootCompleted.config.fechafin))) {
                SimpleDateFormat format = new SimpleDateFormat("HH:mm:ss");
                String f = format.format(dt);
                Log.d("AppDolor", "Hora actual: " + f);
                Utils.obtenerPosicionContador(f);
                Log.d("AppDolor", "Posicion contador: "+
                Receiver_BootCompleted.config.contador);

                // poner alarma
                Receiver_BootCompleted.demanda = false;
                String fecha =
                Utils.ponerAlarma(getApplicationContext());
                Receiver_BootCompleted.var.fecha = fecha;
                Utils.UpdateResultsXML("datos");
            }

            CambioHora.this.finish();
        }

        if (origen==2)
        {
            Intent intent=new Intent(CambioHora.this,VisorPregunta.class);
            intent.putExtra("cont",Receiver_BootCompleted.config.contador );
            CambioHora.this.startActivity(intent);
            CambioHora.this.finish();
        }

    }
}

```

}

}

```
import java.io.File;
import java.lang.reflect.Field;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.os.StrictMode;
import android.app.Activity;
import android.app.DatePickerDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewConfiguration;
import android.view.Window;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.RelativeLayout;
import android.widget.Spinner;
import android.widget.Toast;

public class Conf_inicial extends Activity {

    Button btncontinuar;
    RelativeLayout rl1;
    RelativeLayout rl2;
    ProgressBar pb;
    ProgressBar pb2;
    Context ctx;
    String imei;
    ImageButton calendario;
    EditText fecha;

    private StringBuilder sb = new StringBuilder();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        requestWindowFeature(Window.FEATURE_NO_TITLE);

        // Activity a pantalla completa
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);

        ctx = this;

        setContentView(R.layout.conf_inicial);

        int origen=getIntent().getExtras().getInt("origen",0);
```

```

btncontinuar = (Button) findViewById(R.id.instalacionacceptar);
btncontinuar.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {

        new Wait2().execute("");

    }
});

fecha = (EditText) findViewById(R.id.fecha);
calendario = (ImageButton) findViewById(R.id.calendar);
calendario.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Calendar c = Calendar.getInstance();
        int mYear = c.get(Calendar.YEAR);
        int mMonth = c.get(Calendar.MONTH);
        int mDay = c.get(Calendar.DAY_OF_MONTH);
        System.out.println("the selected " + mDay);
        DatePickerDialog dialog = new DatePickerDialog(Conf_inicial.this, new
        mDateSetListener(), 1970, mMonth, mDay);
        dialog.show();
    }
});

r11 = (RelativeLayout) findViewById(R.id.layoutinstalacion1);
r12 = (RelativeLayout) findViewById(R.id.layoutinstalacion2);
pb = (ProgressBar) findViewById(R.id.pbinstalacion);
pb2 = (ProgressBar) findViewById(R.id.pbinstalacion2);

if (origen==1)
{
    r11.setVisibility(View.VISIBLE);
    r12.setVisibility(View.GONE);
    new Wait().execute("");
}

if (origen==2)
{
    pb.setVisibility(View.GONE);
    r11.setVisibility(View.GONE);
    r12.setVisibility(View.VISIBLE);
    pb2.setVisibility(View.GONE);
}

}

private class Wait extends AsyncTask<String, Void, Integer> {

    protected void onPreExecute() {
        pb.setVisibility(View.VISIBLE);
    }

    protected Integer doInBackground(String... urls) {

        SharedPreferences settings = getApplicationContext()
            .getSharedPreferences(Utils.appname, 0);
        imei = settings.getString("imei", "");
        if (imei.length() == 0) {
            imei = Utils.getIMEI(getApplicationContext());
            Date fechainstalacion = new Date();
            SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
            String fech = format.format(fechainstalacion);
            int r = Utils.almacenarUsuario(imei, fech);
            if (r == 1) {
                SharedPreferences.Editor editor = settings.edit();
                editor.putString("imei", imei);
                editor.commit();
            }
        }
    }
}

```

```

}

// empezar servicios y poner primera alarma
// servicio sms
Intent servicio = new Intent();
servicio.setAction("appcontextualv3.code.SMS");
if (ctx.startService(servicio) == null) {
    Log.d("AppDolor",
        "No se ha podido iniciar el servicio de SMS");
} else {
    Log.d("AppDolor", "Servicio SMS iniciado correctamente");
}

// servicio llamadas se lanza desde manifest

try {
    if (Utils.isOnline(ctx)) {
        //Utils.almacenarResultados();
        Utils.descargarConfInicial("DolorG1");
        Utils.descargarConf();
        Utils.descargarimagenesfichero("confinicial");
        Utils.descargarimagenesfichero("conf");
    }
}

catch (Exception e) {
    Log.d("AppDolor",
        "Error almacenando resultados al iniciar aplicación"
        + e.getMessage());
}

File file = new File(Environment.getExternalStorageDirectory()
    .toString() + Utils.carpetas + "conf.xml");
if (!file.exists()) {
    Toast.makeText(ctx, "No existe fichero de configuración",
        Toast.LENGTH_SHORT).show();
}

else {
    try {
        Log.d("AppDolor", "Parseo fichero configuracion");
        Utils.parsearficheroconfiguracion("conf", ctx);
        Date dt = new Date();

        if ((dt.after(Receiver_BootCompleted.config.fechainicio)
            && (dt.before(Receiver_BootCompleted.config.fechafin)))) {
            SimpleDateFormat format = new SimpleDateFormat(
                "HH:mm:ss");
            String f = format.format(dt);
            Log.d("AppDolor", "Hora actual: " + f);
            Utils.obtenerPosicionContador(f);
            Log.d("AppDolor", "Posicion contador: "
                + Receiver_BootCompleted.config.contador);
            // poner alarma a hora que toque
            String fecha = Utils.ponerAlarma(ctx);
            // limpiar contadores llamadas y mensajes
            Receiver_BootCompleted.var.limpiarVariables();
            boolean b = Utils.existefichero("datos");
            if (b) {
                Utils.parsearfichero("datos");
                Receiver_BootCompleted.var.fecha = fecha;
                Utils.UpdateResultsXML("datos");
            } else {
                SimpleDateFormat ft = new SimpleDateFormat(
                    "yyyy-MM-dd_HH-mm-ss");
                String fi = ft.format(dt);
                Receiver_BootCompleted.var.fecha = fecha;
                Receiver_BootCompleted.var.fecha_inicio = fi;
                Utils.UpdateResultsXML("datos");
            }
        }
    }
}

```

```

        }

        }
    } catch (Exception e) {
        Log.d("AppDolor", "Exception:" + e.getMessage());
    }
}

return null;
}

protected void onPostExecute(Integer valor) {

    pb.setVisibility(View.GONE);
    /*rl1.setVisibility(View.GONE);
    rl2.setVisibility(View.VISIBLE);
    pb2.setVisibility(View.GONE); */

    Intent intent = new Intent(Conf_inicial.this, Instrucciones.class);
    intent.putExtra("origen", 1);
    Conf_inicial.this.startActivity(intent);

    if(Utils.getGPSStatus(ctx) == false){
        String aviso = "Localización por red y GPS desactivado.";
        sb.append(aviso);
        sb.append("\nPor favor active al menos una opción.");
        Toast.makeText(Conf_inicial.this, sb.toString(), Toast.LENGTH_LONG).show();

        Intent settingsIntent = new
        Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
        settingsIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
        Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
        Conf_inicial.this.startActivityForResult(settingsIntent, 0);
    }

    Conf_inicial.this.finish();
}
}

private class Wait2 extends AsyncTask<String, Void, Integer> {

    protected void onPreExecute() {
        pb2.setVisibility(View.VISIBLE);
    }

    protected Integer doInBackground(String... urls) {

        String valordatos=fecha.getText().toString();
        Utils.almacenarDatosUsuario(Receiver_BootCompleted.config.imei, valordatos);

        return null;
    }

    protected void onPostExecute(Integer valor) {

        pb2.setVisibility(View.GONE);
        Intent intent2=new Intent(Conf_inicial.this, VisorPreguntaInicial.class);
        intent2.putExtra("cont", 0 );
        Conf_inicial.this.startActivity(intent2);
        Conf_inicial.this.finish();
    }
}
}

```

```
class mDateSetListener implements DatePickerDialog.OnDateSetListener {

    @Override
    public void onDateSet(DatePicker view, int year, int monthOfYear,
        int dayOfMonth) {
        int mYear = year;
        int mMonth = monthOfYear;
        int mDay = dayOfMonth;
        fecha.setText(new StringBuilder()
            // Month is 0 based so add 1
            .append(mDay).append("/").append(mMonth + 1).append("/")
            .append(mYear).append(" "));
    }
}
```

```
package appcontextualv3.code;
```

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

```
import java.util.Date;
```

```
public class Configuracion{
```

```
    public String imei;
```

```
    public ArrayList<Lanzamiento> lanzamientos=new ArrayList<Lanzamiento>();
```

```
    public int contador=0;
```

```
    public Lanzamiento demanda;
```

```
    public Date fechainicio;
```

```
    public Date fechafin;
```

```
}
```

```
package appcontextualv3.code;
```

```
import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Date;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.os.StrictMode;
import android.util.Log;
import android.view.Window;
import android.view.WindowManager;
import android.widget.ProgressBar;
import android.widget.Toast;
```

```
public class Demanda extends Activity{
```

```
    private ProgressBar pb;
```

```
    @Override
```

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

```
        // Hide the window title.
```

```
        requestWindowFeature(Window.FEATURE_NO_TITLE);
```

```
        // Activity a pantalla completa
```

```
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

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

```
        pb = (ProgressBar) findViewById(R.id.progressBar1);
```

```
        //comprobar si es primera vez y redirigir
```

```
        SharedPreferences settings =
```

```
        getApplicationContext().getSharedPreferences(Utils.appname, 0);
```

```
        String imei=settings.getString("imei", "");
```

```
        if (imei.length()==0)
```

```
        {
```

```
            Intent intent=new Intent(Demanda.this,Conf_inicial.class);
```

```
            intent.putExtra("origen",1 );
```

```
            Demanda.this.startActivity(intent);
```

```
            Demanda.this.finish();
```

```
        }
```

```
    else
```

```
    {
```

```
        boolean datosiniciales=settings.getBoolean("inicial", false);
```

```
        if (!datosiniciales)
```

```
        {
```

```
            Intent intent=new Intent(Demanda.this,Conf_inicial.class);
```

```
            intent.putExtra("origen",2 );
```

```
            Demanda.this.startActivity(intent);
```

```
            Demanda.this.finish();
```

```
        }
```

```
    else
```

```
    {
```

```
        boolean noti=settings.getBoolean("notificacion", false);
```

```
        if (noti)
```

```
        {
```

```
            int contador=settings.getInt("contador", 0);
```

```

Utils.cancelarNotificacion(getApplicationContext());
Intent intent=new Intent(Demanda.this,VisorPregunta.class);
//intent.putExtra("cont",Receiver_BootCompleted.config.contador );
intent.putExtra("cont",contador );
Demanda.this.startActivity(intent);
}

else
{

//si no es la primera vez estoy lanzando a demanda normal
Receiver_BootCompleted.demanda=true;
File file = new
File(Environment.getExternalStorageDirectory().toString()+ Utils.carpeta
+ "conf.xml");
if(!file.exists())
{
    Toast.makeText(this, "No existe fichero de configuración. Por favor
reincia el móvil y asegurate de tener internet",
    Toast.LENGTH_LONG).show();

}
else
{

Utils.parsearficheroconfiguracion("conf",this(getApplicationContext())
;
Utils.parsearfichero("datos");
Date dt= new Date();
Lanzamiento l=Receiver_BootCompleted.config.demanda;
if (l==null)
{
    //Toast.makeText(this, "No tienes habilitada la opción de
contestar a demanda", Toast.LENGTH_LONG).show();
Receiver_BootCompleted.demanda=false;
Intent intent = new Intent(Demanda.this,Instrucciones.class);
intent.putExtra("origen",3);
Demanda.this.startActivity(intent);

}

else
{
    if ((dt.after(l.inicio)) && (dt.before(l.fin)))
    {
        SimpleDateFormat ft = new
SimpleDateFormat("yyyy-MM-dd_HH-mm-ss");
String fi = ft.format(dt);
Receiver_BootCompleted.var.fecha=fi;
Utils.UpdateResultsXML("datos");

//generar el fichero de datos y clima
Receiver_BootCompleted.var .nombrefichero
="results1"+Receiver_BootCompleted.var.fecha;
boolean b= Utils.existefichero(Receiver_BootCompleted.var
.nombrefichero);
if (!b)
{
    Utils.creaFichero1(Receiver_BootCompleted.var
.nombrefichero);
//limpiar las variables que ya he copiado en el fichero
String nuevafechainicio=Receiver_BootCompleted.var.fecha;
Receiver_BootCompleted.var.limpiarVariablesMovil();

Receiver_BootCompleted.var.fecha_inicio=nuevafechainicio;

Utils.UpdateResultsXML("datos");
}
Handler handler = new Handler();
handler.post(new Runnable() {
    public void run() {

```

```
        Utils.comprobarTiempo(getApplicationContext());
    }
    });

    //pasar las preguntas
    Demanda.this.startActivity(new
    Intent(Demanda.this, VisorPregunta.class));
}
else
{
    Toast.makeText(this, "No tienes la opción a demanda
    habilitada en estas fechas", Toast.LENGTH_LONG).show();
    Receiver_BootCompleted.demanda=false;
}
}
}
}
}
Demanda.this.finish();
```

```
}
}
}
```

```
package appcontextualv3.code;

import java.text.SimpleDateFormat;
import java.util.Date;
import android.app.Activity;
import android.content.Context;
import android.graphics.Color;
import android.graphics.Typeface;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.util.Log;
import android.util.TypedValue;
import android.view.Gravity;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.view.ViewGroup.LayoutParams;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.RelativeLayout;
import android.widget.TextView;

public class FinPreguntas extends Activity{

    public ProgressBar pb;

    LocationManager manager = null;
    LocationListener listener = null;
    double latitude;
    double longitude;
    Context ctx;
    int MILISEGUNDOS_ESPERA = 1000*60*4;
    boolean tiempoCalculado=false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setContentView(R.layout.fondo);

        ctx=this;

        LinearLayout fondo=(LinearLayout)findViewById(R.id.layoutfondo);

        RelativeLayout rl= new RelativeLayout(this);
        rl.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
            LayoutParams.MATCH_PARENT));
        rl.setPadding(5, 5, 5, 5);

        TextView tv=new TextView(this);
        tv.setText("Muchas gracias por tu colaboración");
        tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,24);
        tv.setTextColor(Color.parseColor("#2C88BF"));
        tv.setGravity(Gravity.CENTER);
        RelativeLayout.LayoutParams params= new
        RelativeLayout.LayoutParams(LayoutParams.WRAP_CONTENT , LayoutParams.WRAP_CONTENT);
        params.addRule(RelativeLayout.CENTER_HORIZONTAL);
        tv.setLayoutParams(params);
        rl.addView(tv);
```

```

pb=new ProgressBar(this);
pb.getIndeterminateDrawable().setColorFilter(Color.parseColor("#2C88BF"),
android.graphics.PorterDuff.Mode.MULTIPLY);
RelativeLayout.LayoutParams params2= new
RelativeLayout.LayoutParams(LayoutParams.WRAP_CONTENT , LayoutParams.WRAP_CONTENT);
params2.addRule(RelativeLayout.CENTER_IN_PARENT);
pb.setLayoutParams(params2);
rl.addView(pb);

fondo.addView(rl);

pb.setVisibility(View.VISIBLE );

Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    public void run() {

        guardardatos();
    }
},2000);
}

protected void guardardatos() {

//cerrar fichero resultados
Utils.cerrarFicheroResultados("results2"+Receiver_BootCompleted.var.fecha);
//Utils.borrarFichero("datos");
Receiver_BootCompleted.demanda = false;

if (Utils.isOnline(getApplicationContext()))
{

    //politica de seguridad
    StrictMode.ThreadPolicy policy = new
    StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);

    Utils.almacenarResultados("results2", "2");
    Utils.descargarConf();

    Utils.descargarimagenesfichero("confinicial");
    Utils.descargarimagenesfichero("conf");

    Utils.parsearficheroconfiguracion("conf", ctx);
}

Date dt = new Date();

if ((dt.after(Receiver_BootCompleted.config.fechainicio)) &&
(dt.before(Receiver_BootCompleted.config.fechafin))) {
    SimpleDateFormat format = new SimpleDateFormat("HH:mm:ss");
    String f = format.format(dt);
    Log.d("AppDolor", "Hora actual: " + f);
    Utils.obtenerPosicionContador(f);
    Log.d("AppDolor", "Posicion contador: "+ Receiver_BootCompleted.config.contador);

    // poner alarma
    String fecha = Utils.ponerAlarma(ctx);
    Receiver_BootCompleted.var.fecha = fecha;
    Utils.updateResultsXML("datos");
}

pb.setVisibility(View.GONE );

```

```
FinPreguntas.this.finish();
```

```
}
```

```
}
```

```
package appcontextualv3.code;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;

public class Instrucciones extends AppCompatActivity {

    int origen;
    Button ok;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.Lay
outParams.FLAG_FULLSCREEN);

        //barra

        ActionBar actionBar = getSupportActionBar();
        actionBar.setDisplayShowTitleEnabled(false);
        //actionBar.setDisplayShowHomeEnabled(false);
        actionBar.setBackgroundDrawable(new
ColorDrawable(Color.parseColor("#333333")));

        setContentView(R.layout.instrucciones);
        origen=getIntent().getExtras().getInt("origen",0);

        ok = (Button) findViewById(R.id.instruccionesok);
        ok.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {

                if (origen==1)
                {

                    Intent intent2=new Intent(Instrucciones.this,Conf_inicial.class);
                    intent2.putExtra("origen",2 );
                    Instrucciones.this.startActivity(intent2);
                    Instrucciones.this.finish();
                }

                if (origen==2)
                {
                    Intent intent2=new Intent(Instrucciones.this,VisorPregunta.class);
                    intent2.putExtra("cont",Receiver_BootCompleted.config.contador );
                    Instrucciones.this.startActivity(intent2);
                    Instrucciones.this.finish();
                }

                if (origen==3)
                {
                    Instrucciones.this.finish();
                }
            }
        });
    }
}
```

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.

    if (origen==1)
    {
        getMenuInflater().inflate(R.menu.instrucciones1, menu);
    }

    if (origen==2)
    {
        getMenuInflater().inflate(R.menu.instrucciones2, menu);
    }

    if (origen==3)
    {
        getMenuInflater().inflate(R.menu.instrucciones3, menu);
    }
    return true;
}
}
```

```
@Override
public boolean onOptionsItemSelected ( MenuItem item){
    switch(item.getItemId()){
        case R.id.action_settings :
            Intent intent=new Intent(Instrucciones.this,CambioHora.class);
            intent.putExtra("origen",1 );
            Instrucciones.this.startActivity(intent);
            Instrucciones.this.finish();
            return true;
        case R.id.action_exit :
            Instrucciones.this.finish();
            return true;
        case R.id.action_back :
            Intent intent2=new Intent(Instrucciones.this,VisorPregunta.class);
            intent2.putExtra("cont",Receiver_BootCompleted.config.contador );
            Instrucciones.this.startActivity(intent2);
            Instrucciones.this.finish();
            return true;
    }
    return super.onOptionsItemSelected(item);
}
}
```

```
}
```

```
package appcontextualv3.code;

import java.util.ArrayList;
import java.util.Date;

public class Lanzamiento {

    public Date hora;
    public Date inicio;
    public Date fin;
    public ArrayList<Pregunta> preguntas=new ArrayList<Pregunta>();
    public int contadorpregunta;

}
```

```

package appcontextualv3.code;

import android.app.Service;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.telephony.TelephonyManager;
import android.util.Log;

public class Llamadas extends BroadcastReceiver{

    private static boolean incomingFlag = false;
    private static boolean incoming_accepted=false;

    @Override
    public void onReceive(Context arg0, Intent arg1) {
        String action = arg1.getAction();

        if(action.equals(Intent.ACTION_NEW_OUTGOING_CALL))
        {

            Utils.parsearfichero("datos");
            incomingFlag = false;
            incoming_accepted=false;
            Receiver_BootCompleted.var .llamadasrealizadas =Receiver_BootCompleted.var
            .llamadasrealizadas+1;
            Log.d("AppDolor", "LLlamadas realizadas: "+Receiver_BootCompleted.var
            .llamadasrealizadas);
            Utils.UpdateResultsXML("datos");

        }else if(action.equals(TelephonyManager.ACTION_PHONE_STATE_CHANGED))

        {
            TelephonyManager tm =
            (TelephonyManager) arg0.getSystemService(Service.TELEPHONY_SERVICE);
            switch (tm.getCallState()) {

                case TelephonyManager.CALL_STATE_RINGING:

                    incomingFlag = true;
                    incoming_accepted=false;
                    break;

                case TelephonyManager.CALL_STATE_OFFHOOK:

                    if(incomingFlag){
                        Utils.parsearfichero("datos");
                        incoming_accepted=true;
                        incomingFlag =false;
                        Receiver_BootCompleted.var .llamadasrecibidas =
                        Receiver_BootCompleted.var .llamadasrecibidas+1;
                        Log.d("AppDolor", "LLlamadas recibidas: "+Receiver_BootCompleted.var
                        .llamadasrecibidas );
                        Utils.UpdateResultsXML("datos");
                    }
                    break;

                case TelephonyManager.CALL_STATE_IDLE:
                    if(incomingFlag){
                        if (!incoming_accepted)
                        {
                            Utils.parsearfichero("datos");
                            Receiver_BootCompleted.var .llamadasperdidas =
                            Receiver_BootCompleted.var .llamadasperdidas+1;
                            Log.d("AppDolor", "LLlamadas perdidas: "+
                            Receiver_BootCompleted.var .llamadasperdidas);
                            Utils.UpdateResultsXML("datos");
                        }
                    }
                    break;
            }
        }
    }
}

```

}

}

}

}

```
package appcontextualv3.code;

import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.os.Bundle;

public class MyLocationListener implements LocationListener {

    double latitude;
    double longitude;
    Context context;

    public MyLocationListener(Context context2) {
        context=context2;
    }

    @Override
    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub
        // Update the information in the interface "longitude,latitude"
        if (!Utils.tiempocalculado)
        {
            longitude = location.getLongitude();
            latitude = location.getLatitude();
            if((latitude!=0.0) && (longitude!=0.0))
            {
                Utils.tiempocalculado=true;
                Utils.manager.removeUpdates(Utils.listener);
                ObtenerInfoTiempo t = new ObtenerInfoTiempo();
                t.setContext(context);
                t.execute( latitude, longitude);
            }
        }
    }

    @Override
    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }
}
```

```
package appcontextualv3.code;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import org.apache.http.NameValuePair;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.app.ProgressDialog;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;

public class ObtenerInfoTiempo extends AsyncTask<Double, Void, Tiempo> {

    String response = null;

    double latitude;
    double longitude;

    Tiempo weather = null;
    private Context context = null;

    ProgressDialog progressDialog;

    @Override
    protected void onPreExecute() {
        Log.d("AppContextual", "onPreExecute");
        /*
         * progressDialog = new ProgressDialog(context);
         * progressDialog.setMessage("Guardando Datos...");
         * progressDialog.setCancelable(false); progressDialog.show();
         */
    }

    protected Tiempo doInBackground(Double... params) {

        latitude = params[0];
        longitude = params[1];

        if ((latitude != 0.0) && (longitude != 0.0))
        {
            // Web Service

            HttpURLConnection con = null;
            Log.d("AppContextual", "doInBackground");
            try {
                // Create a new list of name-pair values to use as
```

```

// parameters in the request
List<NameValuePair> pairs = new ArrayList<NameValuePair>();
// Add the name and friend's name

pairs.add(new BasicNameValuePair("APPID",
    "5f3861c9ef74284515436964b574cdbc"));
pairs.add(new BasicNameValuePair("lat", String
    .valueOf(latitude)));
pairs.add(new BasicNameValuePair("lon", String
    .valueOf(longitude)));
pairs.add(new BasicNameValuePair("lang", "sp"));
pairs.add(new BasicNameValuePair("units", "metric"));

// Create a new URL
URL url = new URL(
    "http://api.openweathermap.org/data/2.5/weather"
    + "?"
    + URLEncoder.format(pairs, "utf-8"));
// Get a new connection to the required resource
con = (URLConnection) url.openConnection();
con.setRequestMethod("GET");
// Set the header to ask for JSON replies
con.setRequestProperty("Accept", "application/json");

// Process the response using a BufferedReader
BufferedReader reader = new BufferedReader(
    new InputStreamReader(con.getInputStream()));

StringBuffer buffer = new StringBuffer();
String s = null;
while ((s = reader.readLine()) != null) {
    buffer.append(s);
}

response = buffer.toString();
// try parse the string to a JSON object
try {
    // Generate the jsonObject form respobnse
    JSONObject jsonObject = new JSONObject(response);

    weather = new Tiempo();

    weather.setWName(jsonObject.getString("name"));

    JSONArray array = jsonObject.getJSONArray("weather");

    JSONObject object = array.getJSONObject(0);

    weather.setMain(object.getString("main"));
    weather.setDescription(object.getString("description"));
    weather.setIcon(object.getString("icon"));

    JSONObject object2 = jsonObject.getJSONObject("main");
    weather.setTemp(object2.getString("temp"));
    weather.setTempMin(object2.getString("temp_min"));
    weather.setTempMax(object2.getString("temp_max"));

    weather.setPressure(object2.getString("pressure"));
    weather.setHumidity(object2.getString("humidity"));

    JSONObject object3 = jsonObject.getJSONObject("wind");
    weather.setSpeed(object3.getString("speed"));
    weather.setDeg(object3.getString("deg"));

} catch (JSONException e) {
    Log.e("JSON Parser",
        "Error parsing data " + e.toString());
}
} catch (UnsupportedEncodingException uee) {
    Log.d("Tiempo", uee.getMessage());
} catch (IOException ioe) {

```

```

        Log.d("Tiempo", ioe.getMessage());
    } finally {
        // Release connection
        con.disconnect();
    }
}

return weather;
}

public void onPostExecute(Tiempo weather) {
    super.onPostExecute(weather);
    // progressDialog.cancel();
    if (weather != null) {
        Log.d("AppDolor", "Descripcion: " + weather.description);
        Log.d("AppDolor", "Ciudad: " + weather.wname);
        Log.d("AppDolor", "Temperatura: " + weather.temp);
        Log.d("AppDolor", "Humedad: " + weather.humidity);
        Log.d("AppDolor", "Presión: " + weather.pressure);
        Log.d("AppDolor", "Velocidad-viento: " + weather.speed);
        Utils.escribeTiempoFichero(Receiver_BootCompleted.var .nombrefichero,
            weather.wname,
            weather.description, weather.temp, weather.humidity,
            weather.pressure, weather.speed);
    }
    messageHandler.sendMessage(0);
}

public Context getContext() {
    return context;
}

public void setContext(Context context) {
    this.context = context;
}

/*****
 * HANDLER
 *****/

//
////////////////////////////////////
////////////////////////////////////
private Handler messageHandler = new Handler() {
    public void handleMessage(Message msg) {

        switch (msg.what) {
            case 0:
                // cerrar fichero resultados
                Utils.cerrarFicheroResultados(Receiver_BootCompleted.var .nombrefichero);
                //Utils.borrarFichero("datos");

                try
                {
                    if (Utils.isOnline(context)) {

                        //politica de seguridad
                        StrictMode.ThreadPolicy policy = new
                        StrictMode.ThreadPolicy.Builder().permitAll().build();
                        StrictMode.setThreadPolicy(policy);
                        Utils.almacenarResultados("results1", "1");

                    }

                }

            }
}
}

```

```
    catch (Exception e)
    {
        Log.d("AppDolor", "Error almacenando resultados1 tras obtener tiempo" +
            e.getMessage());
    }
```

```
    break;
```

```
default:
    break;
}
```

```
}
```

```
};
```

```
}
```

```
package appcontextualv3.code;
```

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

```
public class Pregunta {
```

```
    public String id;
```

```
    public String texto;
```

```
    public int numrespuestas;
```

```
    public String tipo;
```

```
    public int tiporespuesta;
```

```
    public ArrayList<Respuesta> respuestas=new ArrayList<Respuesta>();
```

```
}
```

```
package appcontextualv3.code;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Date;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Environment;
import android.os.StrictMode;
import android.util.Log;
import android.widget.Toast;

public class Receiver_BootCompleted extends BroadcastReceiver{

    public static Variables var = new Variables();
    public static Configuracion config=new Configuracion();
    public static boolean demanda=false;
    String imei="";

    @Override
    public void onReceive(Context context, Intent intent) {
        Log.d("AppDolor", "Iniciando servicios AppContextual");

        //comprobar si es primera vez y redirigir
        SharedPreferences settings = context.getSharedPreferences(Utils.appname, 0);
        String imei=settings.getString("imei", "");
        if (imei.length()==0)
        {
            Intent i = new Intent(context, Conf_inicial.class);
            i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            context.startActivity(i);
        }

        else
        {
            // he reiniciado el movil

            //servicio sms
            Intent servicio= new Intent();
            servicio.setAction("appcontextualv3.code.SMS");
            if( context.startService(servicio)== null)
            {
                Log.d("AppDolor", "No se ha podido iniciar el servicio de SMS");
            }
            else
            {
                Log.d("AppDolor", "Servicio SMS iniciado correctamente");
            }

            // servicio llamadas se lanza desde manifest

            try
            {
                if (Utils.isOnline(context))
                {
                    //politica de seguridad
                    StrictMode.ThreadPolicy policy = new
                    StrictMode.ThreadPolicy.Builder().permitAll().build();
                    StrictMode.setThreadPolicy(policy);
                    //Utils.almacenarResultados();
                    Utils.descargarConf();
                }
            }
        }
    }
}
```

```

}

catch(Exception e)
{
    Log.d("AppDolor", "Error almacenando resultados al iniciar
    aplicación"+e.getMessage());
}

File file = new File(Environment.getExternalStorageDirectory().toString()+
Utils.carpeta + "conf.xml");
if(!file.exists())
{
    Toast.makeText(context, "No existe fichero de configuración",
    Toast.LENGTH_SHORT).show();

}

else
{
    try
    {
        Log.d("AppDolor", "Parseo fichero configuracion");
        Utils.parsearficheroconfiguracion("conf", context);
        Date dt= new Date();

        if ((dt.after(config.fechainicio ) ) && (dt.before(config.fechafin)))
        {
            SimpleDateFormat format = new SimpleDateFormat("HH:mm:ss");
            String f = format.format(dt);
            Log.d("AppDolor", "Hora actual: "+f);
            Utils.obtenerPosicionContador(f);
            Log.d("AppDolor", "Posicion contador:
            "+Receiver_BootCompleted.config.contador);
            // poner alarma a hora que toque
            String fecha=Utils.ponerAlarma(context);
            //limpiar contadores llamadas y mensajes
            Receiver_BootCompleted.var.limpiarVariables();
            boolean b=Utils.existefichero("datos");
            if (b)
            {
                Utils.parsearfichero("datos");
                Receiver_BootCompleted.var.fecha=fecha;
                Utils.UpdateResultsXML("datos");
            }
            else
            {
                SimpleDateFormat ft = new SimpleDateFormat("yyyy-MM-dd_HH-mm-ss");
                String fi = ft.format(dt);
                Receiver_BootCompleted.var.fecha=fecha;
                Receiver_BootCompleted.var.fecha_inicio=fi;
                Utils.UpdateResultsXML("datos");
            }

            SharedPreferences.Editor editor = settings.edit();
            editor.putBoolean("notificacion", false);
            editor.putInt("contador", 0);
            editor.commit();

        }
        else
        {

            if (dt.after(config.fechafin))
            {
                Log.d("AppDolor", "Fin periodo AppDolor. Intento desinstalar");
                Uri packageUri = Uri.parse("package:appcontextualv3.code");
                Intent uninstallIntent =new
                Intent(Intent.ACTION_DELETE,packageUri);
                uninstallIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            }
        }
    }
}

```

```
        context.startActivity(uninstallIntent);
    }
}
catch(Exception e)
{
    Log.d("EMA", "Exception:" + e.getMessage());
}
}
}
}
```

```
package appcontextualv3.code;

import java.text.SimpleDateFormat;
import java.util.Date;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Handler;
import android.os.StrictMode;
import android.support.v4.app.NotificationCompat;
import android.util.Log;

public class RecibeAlarma extends BroadcastReceiver{

    private NotificationManager mManager;
    private Context context;

    private void Notificar(Intent intentNot) {

        Notification notification = new NotificationCompat.Builder(context)
            .setContentIntent(PendingIntent.getActivity(context, 0, intentNot,
                PendingIntent.FLAG_CANCEL_CURRENT))
            .setWhen(System.currentTimeMillis())
            .setTicker(context.getResources().getString(R.string.ticker_notificacion))
            .setContentTitle(context.getResources().getString(R.string.app_name))
            .setContentText(context.getResources().getString(R.string.texto_notificacion))
            .setSmallIcon(R.drawable.logo3)
            .build();

        notification.defaults |= Notification.DEFAULT_SOUND;
        notification.defaults |= Notification.DEFAULT_VIBRATE;
        notification.defaults |= Notification.DEFAULT_LIGHTS;
        notification.flags = Notification.FLAG_AUTO_CANCEL;

        try
        {
            notification.ledARGB = 0xff00ff00;
            notification.ledOnMS = 300;
            notification.ledOffMS = 1000;
            notification.flags |= Notification.FLAG_SHOW_LIGHTS;
        }catch (Exception ex)
        {

        }

        mManager.notify(Utils.APP_ID_NOTIFICATION, notification);

    }

    @Override
    public void onReceive(Context arg0, Intent intent) {
        context=arg0;
        mManager = (NotificationManager)
            context.getSystemService(Context.NOTIFICATION_SERVICE);

        Date dt= new Date();
        SimpleDateFormat format = new SimpleDateFormat("HH:mm");
        String f = format.format(dt);
        Utils.parsearchivoconfiguracion("conf", arg0);
        Utils.calcularposicionactualcontador(f);
        int cont=Receiver_BootCompleted.config.contador;
        Log.d("AppDolor", "Recibe alarma posicion contador: "+ cont);
        Utils.parsearchivo("datos");
```

```

//comprobar que es la fecha que toca y rellene la vez anterior
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd_HH-mm-ss");
String fechastring = sdf.format(dt.getTime());
Log.d("AppDolor", "Fecha del fichero: "+ Receiver_BootCompleted.var .fecha);
Log.d("AppDolor", "Fecha real: "+ fechastring);
if (fechastring.compareTo(Receiver_BootCompleted.var .fecha )!=0)
{
    Receiver_BootCompleted.var.fecha=fechastring;
    Utils.UpdateResultsXML("datos");
    Log.d("AppDolor", "Tengo que modificar la fecha del fichero: "+ fechastring);
}

if (cont>-1)
{
    Lanzamiento l=Receiver_BootCompleted.config.lanzamientos.get(cont);
    if ((dt.after(l.inicio)) && (dt.before(l.fin)))
    {
        Receiver_BootCompleted.demanda=false;

        //mandar fichero de datos movil y tiempo
        Utils.parsearfichero("datos");
        Receiver_BootCompleted.var .nombrefichero
        ="results1"+Receiver_BootCompleted.var.fecha;
        boolean b= Utils.existefichero(Receiver_BootCompleted.var .nombrefichero);
        if (!b)
        {
            Utils.creaFichero1(Receiver_BootCompleted.var .nombrefichero);
            //limpiar las variables que ya he copiado en el fichero
            String nuevafechainicio=Receiver_BootCompleted.var.fecha;
            Receiver_BootCompleted.var.limpiarVariablesMovil();
            Receiver_BootCompleted.var.fecha_inicio=nuevafechainicio;
            Utils.UpdateResultsXML("datos");
        }
        Handler handler = new Handler();
        handler.post(new Runnable() {
            public void run() {
                Utils.comprobartiempo(context);
            }
        });

        //poner notificacion
        Intent lanzar=new Intent(context, VisorPregunta.class);
        lanzar.putExtra("cont", cont);
        Notificar(lanzar);
        SharedPreferences settings = context.getSharedPreferences(Utils.appname, 0);
        SharedPreferences.Editor editor = settings.edit();
        editor.putBoolean("notificacion", true);
        editor.putInt("contador", cont);
        editor.commit();

        //ponersiguientealarma- no avanza el contador
        Utils.ponerProximaAlarma(arg0,cont+1);
    }
    else
    {
        //avanzar contador y poner alarma siguiente
        Receiver_BootCompleted.demanda=false;
        Utils.avanzarContador();
        String fecha=Utils.ponerAlarma(arg0);
        Receiver_BootCompleted.var.fecha=fecha;
        Utils.UpdateResultsXML("datos");
    }
}
}

```

}

}

```
package appcontextualv3.code;

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

public class RecibeAlarma2 extends BroadcastReceiver{

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

        Intent i=new Intent("any string");
        i.setClass(context, Receiver_BootCompleted.class);
        context.sendBroadcast(i);

    }

}
```

```
package appcontextualv3.code;
```

```
public class Respuesta {  
    public String texto;  
    public String imagen;  
    public int valor;  
}
```

```
package appcontextualv3.code;

import android.app.Service;
import android.content.ContentResolver;
import android.content.Intent;
import android.database.ContentObserver;
import android.database.Cursor;
import android.net.Uri;
import android.os.Handler;
import android.os.IBinder;
import android.util.Log;

public class SMS extends Service{

    SMSObserver myObserver=null;
    int lastId=0;

    @Override
    public IBinder onBind(Intent arg0) {
        return null;
    }

    @Override
    public void onCreate()
    {
        super.onCreate();
        Handler handler = new Handler();
        if (myObserver==null)
        {
            myObserver = new SMSObserver(handler);
            ContentResolver contentResolver = getContentResolver();
            contentResolver.registerContentObserver(Uri.parse("content://sms"), true,
            myObserver);
            lastId=0;
        }

    }

    @Override
    public void onDestroy()
    {
        if (myObserver != null) {
            getContentResolver().unregisterContentObserver(myObserver);
        }
        super.onDestroy();
    }

    private class SMSObserver extends ContentObserver{

        public SMSObserver(Handler handler) {
            super(handler);
        }

        public void onChange(boolean selfChange) {
            super.onChange(selfChange);
            Uri uriSMSURI = Uri.parse("content://sms");
            Cursor cur = getContentResolver().query(uriSMSURI, null, null,null, null);
            cur.moveToNext();
            int id = Integer.parseInt(cur.getString(cur.getColumnIndex("_id")));
            Log.d("AppDolor", "Mensaje Id: "+id);
            if (lastId == id)
            {
                return;
            }
            lastId=id;
        }
    }
}
```

```
String tipo=cur.getString(cur.getColumnIndexOrThrow("type"));
String protocol = cur.getString(cur.getColumnIndex("protocol"));
int type = Integer.valueOf(tipo).intValue();
//if(protocol==null && (type == 2 || type==4)){
if(protocol==null && type==4){
    //MESSAGE_TYPE_SENT = 2;
    //MESSAGE_TYPE_OUTBOX = 4;
    Utils.parsearchivo("datos");
    Receiver_BootCompleted.var .mensajesenviados= Receiver_BootCompleted.var
    .mensajesenviados+1;
    Log.d("AppDolor", "Mensajes enviados: "+Receiver_BootCompleted.var
    .mensajesenviados);
    Utils.UpdateResultsXML("datos");
}
if (protocol!=null && type==1){
    //MESSAGE_TYPE_INBOX = 1;
    Utils.parsearchivo("datos");
    Receiver_BootCompleted.var .mensajesrecibidos= Receiver_BootCompleted.var
    .mensajesrecibidos+1;
    Log.d("AppDolor", "Mensajes recibidos: "+ Receiver_BootCompleted.var
    .mensajesrecibidos);
    Utils.UpdateResultsXML("datos");
}
cur.close();
}
}
}
```

```
package appcontextualv3.code;
```

```
public class Tiempo {
```

```
    protected double log;
```

```
    protected double lat;
```

```
    //Main
```

```
    protected String temp = null;
```

```
    protected String pressure = null;
```

```
    protected String humidity = null;
```

```
    protected String temp_min = null;
```

```
    protected String temp_max = null;
```

```
    //Wind
```

```
    protected String speed = null;
```

```
    protected String deg = null;
```

```
    //Weather
```

```
    protected int id;
```

```
    protected String main = null;
```

```
    protected String description = null;
```

```
    protected String icon = null;
```

```
    protected String wname = null;
```

```
    public String getMain() {
```

```
        return main;
```

```
    }
```

```
    public void setMain(String main) {
```

```
        this.main = main;
```

```
    }
```

```
    public String getDescription() {
```

```
        return description;
```

```
    }
```

```
    public void setDescription(String description) {
```

```
        this.description = description;
```

```
    }
```

```
    public String getIcon() {
```

```
        return icon;
```

```
    }
```

```
    public void setIcon(String icon) {
```

```
        this.icon = icon;
```

```
    }
```

```
    public String getWName() {
```

```
        return wname;
```

```
    }
```

```
    public void setWName(String wname) {
```

```
        this.wname = wname;
```

```
    }
```

```
    public String getTemp() {
```

```
        return temp;
```

```
    }
```

```
    public void setTemp(String temp) {
```

```
        this.temp = temp;
```

```
    }
```

```
    public String getTempMin() {
```

```
        return temp_min;
```

```
    }
```

```
    public void setTempMin(String temp_min) {
```

```
        this.temp_min = temp_min;
```

```
}
public String getTempMax() {
    return temp_max;
}

public void setTempMax(String temp_max) {
    this.temp_max = temp_max;
}

public String getSpeed() {
    return speed;
}

public void setSpeed(String speed) {
    this.speed = speed;
}

public String getDeg() {
    return deg;
}

public void setDeg(String deg) {
    this.deg = deg;
}

public String getPressure() {
    return pressure;
}

public void setPressure(String pressure) {
    this.pressure = pressure;
}

public String getHumidity() {
    return humidity;
}

public void setHumidity(String humidity) {
    this.humidity = humidity;
}
}
```

```
package appcontextualv3.code;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.SocketTimeoutException;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Locale;
import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapPrimitive;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserFactory;
import android.app.AlarmManager;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.Drawable;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.ParseException;
import android.os.Environment;
import android.os.Handler;
import android.provider.Settings;
import android.telephony.TelephonyManager;
import android.util.Log;

public class Utils {

    static LocationManager manager = null;
    static LocationListener listener = null;
    static double latitude;
    static double longitude;
    static int MILISEGUNDOS_ESPERA = 1000*60*4;
    static boolean tiempocalculado=false;

    public static final int APP_ID_NOTIFICATION = 0;
    private static final String namespace = "http://tempuri.org/";
    // private static final String url = "http://158.42.170.54:9123/Service.asmx";
    private static final String url = "http://appcontextual.labsitec.es/Service.asmx";
    private static final int timeout = 5 * 1000 * 60;
    public static String appname="AppDolor";
    public static String carpeta="/" + appname + "/";
    public static String
    direccionImagenes="http://appcontextual.labsitec.es/ImagenesPreguntas/";

    public static String getIMEI(Context ctx){
        TelephonyManager telephony = (TelephonyManager)
        ctx.getSystemService(Context.TELEPHONY_SERVICE);
        return telephony.getDeviceId();
    }
}
```

```

public static boolean isOnline(Context ctx) {
    ConnectivityManager cm = (ConnectivityManager)
    ctx.getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo netInfo = cm.getActiveNetworkInfo();
    if (netInfo != null && netInfo.isConnectedOrConnecting()) {
        return true;
    }
    return false;
}

public static void almacenarResultados(String cadena, String tipo){
    String[] fileList = null;
    String xml = null;
    try
    {
        File directorio = new File(Environment.getExternalStorageDirectory().toString()+
        carpeta);
        if(directorio.exists())
        {
            fileList = directorio.listFiles();
            if(fileList.length > 0)
            {
                for(int i=0; i<fileList.length; i++)
                {
                    String fich = fileList[i];
                    if(fich.contains(cadena) == true)
                    {
                        File file = new
                        File(Environment.getExternalStorageDirectory().toString()+
                        carpeta + fich);
                        if(file.exists()){
                            BufferedReader fin = new BufferedReader(new
                            InputStreamReader(new FileInputStream(file)));
                            xml = fin.readLine();
                            if (xml==null)
                            {
                                file.delete();
                            }
                            else
                            {
                                int b=subirficheroresultados(xml, tipo);
                                //if ((b==1) || (b==0))
                                if (b==1)
                                {
                                    file.delete();
                                }

                                xml = null;
                            }
                        }
                    }
                }
            }
        }
    }
    catch (Exception ex){
        Log.e("AppDolor", "Error accediendo a ficheros de resultados");
    }
}

public static int subirficheroresultados(String xml, String tipo)
{
    String Metodo = "AlmacenarResultadosAppPorTipo";
    String accionSoap = namespace + Metodo;

```

```
try {
    // Modelo el request
    SoapObject request = new SoapObject(namespace, Metodo);
    request.addProperty("resultados", xml);
    request.addProperty("tipo", tipo);

    // Modelo el Sobre
    SoapSerializationEnvelope sobre = new
    SoapSerializationEnvelope(SoapEnvelope.VER11);
    sobre.dotNet = true;
    sobre.setOutputSoapObject(request);

    // Modelo el transporte
    HttpTransportSE transporte = new HttpTransportSE(url, timeout);

    // Llamada
    transporte.debug = true;
    transporte.call(accionSoap, sobre);

    // Resultado

    SoapPrimitive resultado = (SoapPrimitive) sobre.getResponse();
    Log.d("AppDolor", "RESULTADO: " + resultado.toString());
    if (resultado.toString().equals("1"))
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

catch (Exception e)
{
    Log.d("AppDolor", "Error almacenando resultado");
    return -1;
}
}

public static int almacenarUsuario(String imei, String fecha)
{
    String Metodo = "AlmacenarUsuarioAppConFecha";
    String accionSoap = namespace + Metodo;

    try {
        // Modelo el request
        SoapObject request = new SoapObject(namespace, Metodo);
        request.addProperty("imei", imei);
        request.addProperty("app", appname);
        request.addProperty("fecha", fecha);

        // Modelo el Sobre
        SoapSerializationEnvelope sobre = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
        sobre.dotNet = true;
        sobre.setOutputSoapObject(request);

        // Modelo el transporte
        HttpTransportSE transporte = new HttpTransportSE(url, timeout);

        // Llamada
        transporte.debug = true;
        transporte.call(accionSoap, sobre);

        // Resultado
```

```

        SoapPrimitive resultado = (SoapPrimitive) sobre.getResponse();
        Log.d("AppDolor", "RESULTADO: " + resultado.toString());
        if (resultado.toString().equals("1"))
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }

    catch (Exception e)
    {
        Log.d("AppContextual", "Error almacenando resultado");
        return -1;
    }
}

public static int almacenarDatosUsuario(String imei, String datos)
{
    String Metodo = "AlmacenarDatosUsuarioApp";
    String accionSoap = namespace + Metodo;

    try {
        // Modelo el request
        SoapObject request = new SoapObject(namespace, Metodo);
        request.addProperty("imei", imei);
        request.addProperty("app", appname);
        request.addProperty("datos", datos);

        // Modelo el Sobre
        SoapSerializationEnvelope sobre = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
        sobre.dotNet = true;
        sobre.setOutputSoapObject(request);

        // Modelo el transporte
        HttpTransportSE transporte = new HttpTransportSE(url, timeout);

        // Llamada
        transporte.debug = true;
        transporte.call(accionSoap, sobre);

        // Resultado

        SoapPrimitive resultado = (SoapPrimitive) sobre.getResponse();
        Log.d("AppDolor", "RESULTADO: " + resultado.toString());
        if (resultado.toString().equals("1"))
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }

    catch (Exception e)
    {
        Log.d("AppDolor", "Error almacenando resultado");
        return -1;
    }
}

```

```
public static int almacenarObservacionesUsuario(String imei, String observaciones)
{
    String Metodo = "AlmacenarObservacionesUsuarioApp";
    String accionSoap = namespace + Metodo;

    try {
        // Modelo el request
        SoapObject request = new SoapObject(namespace, Metodo);
        request.addProperty("imei", imei);
        request.addProperty("app", appname);
        request.addProperty("observaciones", observaciones);

        // Modelo el Sobre
        SoapSerializationEnvelope sobre = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
        sobre.dotNet = true;
        sobre.setOutputSoapObject(request);

        // Modelo el transporte
        HttpTransportSE transporte = new HttpTransportSE(url, timeout);

        // Llamada
        transporte.debug = true;
        transporte.call(accionSoap, sobre);

        // Resultado

        SoapPrimitive resultado = (SoapPrimitive) sobre.getResponse();
        Log.d("AppDolor", "RESULTADO: " + resultado.toString());
        if (resultado.toString().equals("1"))
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }

    catch (Exception e)
    {
        Log.d("AppDolor", "Error almacenando resultado");
        return -1;
    }
}

public static void descargarConf() {
    String Metodo = "ObtenerConfiguracionApp";
    String accionSoap = namespace + Metodo;

    try {
        SoapObject request = new SoapObject(namespace, Metodo);
        // Modelo el Sobre
        SoapSerializationEnvelope sobre = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
        request.addProperty("app", appname);

        sobre.dotNet = true;
        sobre.setOutputSoapObject(request);

        // Modelo el transporte
        HttpTransportSE transporte = new HttpTransportSE(url, timeout);

        // Llamada
```

```

transporte.call(accionSoap, sobre);

// Resultado
SoapPrimitive resultado = (SoapPrimitive) sobre.getResponse();

String xml = resultado.toString();

toXML(xml, "conf");

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

}

public static void descargarConfInicial(String conjuntoinicial) {
    String Metodo = "ObtenerConfiguracionInicialApp";
    String accionSoap = namespace + Metodo;

    try {
        SoapObject request = new SoapObject(namespace, Metodo);
        // Modelo el Sobre
        SoapSerializationEnvelope sobre = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
        request.addProperty("nombreconjuntoinicial", conjuntoinicial);

        sobre.dotNet = true;
        sobre.setOutputSoapObject(request);

        // Modelo el transporte
        HttpTransportSE transporte = new HttpTransportSE(url, timeout);

        // Llamada
        transporte.call(accionSoap, sobre);

        // Resultado
        SoapPrimitive resultado = (SoapPrimitive) sobre.getResponse();

        String xml = resultado.toString();

        toXML(xml, "confinicial");

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

}

private static void toXML(String xml, String nombreFichero) {
    try
    {
        String extStore = Environment.getExternalStorageDirectory().toString();
        File ff=new File(extStore +carpeta);
        if (!ff.exists())
        {
            ff.mkdirs();
        }

        File f = new File(extStore+ carpeta + nombreFichero + ".xml");

        OutputStreamWriter fout = new OutputStreamWriter(new FileOutputStream(f));

        fout.write(xml);
    }
}

```

```

        fout.flush();
        fout.close();
    } catch (Exception ex) {
        Log.e("AppDolor", "Error al escribir fichero a tarjeta SD");
    }
}

public static void UpdateResultsXML(String nombreFichero)
{
    try
    {
        String log = null;
        String extStore = Environment.getExternalStorageDirectory().toString();
        File file = new File(extStore +carpeta+ nombreFichero + ".xml");
        if(file.exists())
        {
            file.delete();
        }

        File ff=new File(extStore +carpeta);
        if (!ff.exists())
        {
            ff.mkdirs();
        }

        File f = new File(extStore +carpeta+ nombreFichero + ".xml");
        f.createNewFile();
        log = "<?xml version=\"1.0\" encoding=\"UTF-8\"?><config>";
        log= log + "<fecha>" + Receiver_BootCompleted.var.fecha + "</fecha>";
        log= log + "<fechainicio>" + Receiver_BootCompleted.var.fecha_inicio +
            "</fechainicio>";
        log= log + "<llamadasperdidas>" + Receiver_BootCompleted.var.llamadasperdidas +
            "</llamadasperdidas>";
        log= log + "<llamadasrecibidas>" + Receiver_BootCompleted.var.llamadasrecibidas +
            "</llamadasrecibidas>";
        log= log + "<llamadasrealizadas>" + Receiver_BootCompleted.var.llamadasrealizadas +
            "</llamadasrealizadas>";
        log= log + "<mensajesenviados>" + Receiver_BootCompleted.var.mensajesenviados +
            "</mensajesenviados>";
        log= log + "<mensajesrecibidos>" + Receiver_BootCompleted.var.mensajesrecibidos +
            "</mensajesrecibidos>";
        log= log+ "</config>";
        OutputStreamWriter fout =new OutputStreamWriter(new FileOutputStream(f));
        fout.write(log);
        fout.flush();
        fout.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

public static void parsearfichero(String nombrefichero)
{
    String extStore = Environment.getExternalStorageDirectory().toString();
    File file = new File(extStore +carpeta+ nombrefichero + ".xml");
    if(file.exists())
    {
        try {
            XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
            XmlPullParser xpp = factory.newPullParser();
            FileInputStream fis = new FileInputStream(file);
            xpp.setInput(new InputStreamReader(fis));
            int event = xpp.getEventType();
            while (event != XmlPullParser.END_DOCUMENT) {
                String name=xpp.getName();
                switch (event){
                    case XmlPullParser.START_TAG:
                        if (name.equals("fecha"))
                        {

```

```

        Receiver_BootCompleted.var.fecha =xpp.nextText();
    }

    if (name.equals("fechainicio"))
    {
        Receiver_BootCompleted.var.fecha_inicio =xpp.nextText();
    }
    if (name.equals("llamadasperdidas"))
    {
        Receiver_BootCompleted.var.llamadasperdidas
        =Integer.parseInt(xpp.nextText());
    }
    if (name.equals("llamadasrecibidas"))
    {
        Receiver_BootCompleted.var.llamadasrecibidas
        =Integer.parseInt(xpp.nextText());
    }
    if (name.equals("llamadasrealizadas"))
    {
        Receiver_BootCompleted.var.llamadasrealizadas
        =Integer.parseInt(xpp.nextText());
    }
    if (name.equals("mensajesenviados"))
    {
        Receiver_BootCompleted.var.mensajesenviados
        =Integer.parseInt(xpp.nextText());
    }
    if (name.equals("mensajesrecibidos"))
    {
        Receiver_BootCompleted.var.mensajesrecibidos
        =Integer.parseInt(xpp.nextText());
    }

    break;

    case XmlPullParser.END_TAG:

    break;
}

    event = xpp.next();
}

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

}

public static void parsearficheroconfiguracion(String nombrefichero, Context ctx) {
    String extStore = Environment.getExternalStorageDirectory().toString();
    File file = new File(extStore +carpeta+ nombrefichero + ".xml");
    if(file.exists())
    {
        try {

            //obtener fecha inicio
            SharedPreferences settings = ctx.getSharedPreferences(appname, 0);
            String fechainicio=settings.getString("fecha_inicio", "");
            Calendar cal=Calendar.getInstance();
            SimpleDateFormat ft = new SimpleDateFormat("dd/MM/yyyy");

            if (fechainicio.length()==0)
            {
                fechainicio = ft.format(cal.getTime());
                SharedPreferences.Editor editor = settings.edit();

```

```

        editor.putString("fecha_inicio", fechainicio);
        editor.commit();
    }

    //parsear datos
    Receiver_BootCompleted .config= new Configuracion();
    String imei=settings.getString("imei", "");
    Receiver_BootCompleted .config.imei =imei;

    Lanzamiento l=new Lanzamiento();
    Pregunta p=new Pregunta();
    Respuesta r=new Respuesta();
    XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
    XmlPullParser xpp = factory.newPullParser();
    FileInputStream fis = new FileInputStream(file);
    xpp.setInput(new InputStreamReader(fis));
    int event = xpp.getEventType();
    while (event != XmlPullParser.END_DOCUMENT) {
        String name=xpp.getName();
        switch (event){
            case XmlPullParser.START_TAG:

                if (name.equals("maxduracion"))
                {

                    int duracion=Integer.parseInt(xpp.nextText());
                    try {
                        Receiver_BootCompleted.config.fechainicio =
                            ft.parse(fechainicio);
                        cal.setTime(ft.parse(fechainicio));
                    } catch (ParseException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                    cal.add(Calendar.DAY_OF_MONTH , duracion);
                    cal.set(Calendar.HOUR_OF_DAY , 23);
                    cal.set(Calendar.MINUTE , 59);
                    cal.set(Calendar.SECOND ,59);

                    Receiver_BootCompleted.config.fechafin =cal.getTime();
                }

                if (name.equals("lanzar"))
                {
                    l=new Lanzamiento();
                }

                if (name.equals("demanda"))
                {
                    l=new Lanzamiento();
                }
                if (name.equals("hora"))
                {

                    String hora=xpp.nextText();
                    SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm:ss");
                    Date convertedDate = new Date();
                    try {
                        convertedDate = dateFormat.parse(hora);
                    } catch (ParseException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }
                l.hora =convertedDate;
            }

            if (name.equals("duracion"))
            {

```

```

        int duracion=Integer.parseInt(xpp.nextText());
        try {
            l.inicio = ft.parse(fechainicio);
            cal.setTime(ft.parse(fechainicio));
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        cal.add(Calendar.DAY_OF_MONTH , duracion);
        cal.set(Calendar.HOUR_OF_DAY , 23);
        cal.set(Calendar.MINUTE , 59);
        cal.set(Calendar.SECOND ,59);

        l.fin =cal.getTime();

    }

    if (name.equals("pregunta"))
    {
        p=new Pregunta();
    }
    if (name.equals("id"))
    {
        p.id =xpp.nextText();
    }

    if (name.equals("texto"))
    {
        p.texto=xpp.nextText();
    }
    if (name.equals("numeroRespuestas"))
    {
        p.numrespuestas =Integer.parseInt(xpp.nextText());
    }
    if (name.equals("tiporespuesta"))
    {
        p.tiporespuesta =Integer.parseInt(xpp.nextText());
    }

    if (name.equals("tipo"))
    {
        p.tipo=xpp.nextText();
    }
    if (name.equals("respuesta"))
    {
        r=new Respuesta();
    }
    if (name.equals("textorespuesta"))
    {
        r.texto=xpp.nextText();
    }
    if (name.equals("imagenrespuesta"))
    {
        r.imagen=xpp.nextText();
    }

    if (name.equals("valor"))
    {
        r.valor =Integer.parseInt(xpp.nextText());
    }

    break;

    case XmlPullParser.END_TAG:
        if (name.equals("lanzar"))
        {
            Receiver_BootCompleted .config.lanzamientos.add(l);
        }
        if (name.equals("demanda"))

```

```

        {
            Receiver_BootCompleted .config.demanda=1;
        }

        if (name.equals("pregunta"))
        {
            l.preguntas .add(p);
        }

        if (name.equals("respuesta"))
        {
            p.respuestas .add(r);
        }
        break;
    }
}

event = xpp.next();
}

// una vez parseado cambio las horas en funcion de lo que hay guardado en
configuracion

for (int j=0 ; j<Receiver_BootCompleted .config.lanzamientos.size();j++)
{
    SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm:ss");
    String horaorigen = dateFormat.format(Receiver_BootCompleted
        .config.lanzamientos.get(j). hora);
    SharedPreferences.Editor editor = settings.edit();
    editor.putString("hora"+j,horaorigen);
    editor.commit();

    Calendar cal2=Calendar.getInstance();
    cal2.setTime(Receiver_BootCompleted .config.lanzamientos.get(j). hora);
    int opcionhora=settings.getInt("opcionhora"+j, 2);
    switch (opcionhora) {
        case 0:
            cal2.add(Calendar.HOUR_OF_DAY, -2);
            break;
        case 1:
            cal2.add(Calendar.HOUR_OF_DAY, -1);
            break;
        case 3:
            cal2.add(Calendar.HOUR_OF_DAY, 1);
            break;
        case 4:
            cal2.add(Calendar.HOUR_OF_DAY, 2);
            break;
        default:
            break;
    }
    Receiver_BootCompleted.config.lanzamientos.get(j).hora
    =cal2.getTime();
}

int opciondias=settings.getInt("opciondias", 2);
int duracion=0;
switch (opciondias) {
    case 0:
        duracion=26;
        break;
    case 1:
        duracion=28;
        break;
    case 2:
        duracion=30;
        break;
    default:
        break;
}
}

```



```

        {
            p.texto=xpp.nextText();
        }
        if (name.equals("numeroRespuestas"))
        {
            p.numrespuestas =Integer.parseInt(xpp.nextText());
        }
        if (name.equals("tiporespuesta"))
        {
            p.tiporespuesta =Integer.parseInt(xpp.nextText());
        }

        if (name.equals("tipo"))
        {
            p.tipo=xpp.nextText();
        }
        if (name.equals("respuesta"))
        {
            r=new Respuesta();
        }
        if (name.equals("textorespuesta"))
        {
            r.texto=xpp.nextText();
        }
        if (name.equals("imagenrespuesta"))
        {
            r.imagen=xpp.nextText();
        }
        if (name.equals("valor"))
        {
            r.valor =Integer.parseInt(xpp.nextText());
        }

        break;

    case XmlPullParser.END_TAG:
        if (name.equals("pregunta"))
        {
            l.preguntas .add(p);
        }

        if (name.equals("respuesta"))
        {
            p.respuestas .add(r);
        }
        break;
    }

    event = xpp.next();
}

} catch (Exception ex)
{
    ex.printStackTrace();
}
}
return l;
}

public static void obtenerPosicionContador(String f) {

    boolean encontrado=false;
    int i=0;

    while ((!encontrado)&& (i<Receiver_BootCompleted.config .lanzamientos.size()))
    {
        Date fechalanzamiento=Receiver_BootCompleted.config .lanzamientos.get(i).hora;
        SimpleDateFormat format = new SimpleDateFormat("HH:mm:ss");

```

```
Date fechaactual= new Date();
try
{
    fechaactual = format.parse(f);

} catch (java.text.ParseException e) {
    e.printStackTrace();
}

if (fechalanzamiento.after(fechaactual))
{
    encontrado=true;
    Receiver_BootCompleted.config.contador=i;

}
i=i+1;

}

if (!encontrado)
{
    Receiver_BootCompleted.config.contador=-1;
}

}
```

```
public static void calcularposicionactualcontador(String f) {
```

```
    boolean encontrado=false;
    int i=0;
```

```
    while ((!encontrado)&& (i<Receiver_BootCompleted.config.lanzamientos.size()))
    {
        Date fechalanzamiento=Receiver_BootCompleted.config.lanzamientos.get(i).hora;
        SimpleDateFormat format = new SimpleDateFormat("HH:mm");
        Date fechaactual= new Date();
        try
        {
            fechaactual = format.parse(f);

        } catch (java.text.ParseException e) {
            e.printStackTrace();
        }

        if (fechalanzamiento.compareTo(fechaactual)==0)
        {
            encontrado=true;
            Receiver_BootCompleted.config.contador=i;

        }
        i=i+1;

    }

    if (!encontrado)
    {
        Receiver_BootCompleted.config.contador=-1;
    }

}
```

```
public static void avanzarContador() {
    int contador=Receiver_BootCompleted.config.contador+1;
    if (contador<Receiver_BootCompleted.config.lanzamientos.size())
    {
        Receiver_BootCompleted.config.contador=contador;
    }
    else
    {
```

```

Receiver_BootCompleted.config.contador=-1;
}
}

public static boolean existefichero(String nombrefichero) {

    try
    {
        String extStore = Environment.getExternalStorageDirectory().toString();
        File file = new File(extStore +carpeta+ nombrefichero + ".xml");
        if(file.exists())
        {
            return true;
        }
    }
    catch(Exception e)
    {
        return false;
    }

    return false;
}

public static void renombrarfichero(String fecha, String fi)
{
    try
    {
        String extStore = Environment.getExternalStorageDirectory().toString();
        File file = new File(extStore +carpeta+ fecha + ".xml");
        File file2 = new File(extStore +carpeta+ fi + ".xml");
        file.renameTo(file2);

    }
    catch(Exception e)
    {
        Log.d("AppContextual", "Error renombrando fichero: " + fecha);
    }

}

public static void creaFichero1(String nombrefichero) {
    try
    {
        String log = null;
        String extStore = Environment.getExternalStorageDirectory().toString();
        File file = new File(extStore +carpeta+ nombrefichero + ".xml");
        if(!file.exists())
        {
            File f = new File(extStore +carpeta+ nombrefichero + ".xml");
            f.createNewFile();
            log = "<?xml version=\"1.0\" encoding=\"UTF-8\"?><results>";
            log= log + "<imei>"+Receiver_BootCompleted.config.imei + "</imei>";
            if (Receiver_BootCompleted.demanda)
            {
                log=log + "<demanda>1</demanda>";
            }
            else
            {
                log=log + "<demanda>0</demanda>";
            }

            log= log + "<fechainicio>"+Receiver_BootCompleted.var.fecha_inicio +
            "</fechainicio>";
            log= log + "<fecha>"+Receiver_BootCompleted.var.fecha + "</fecha>";
            log= log
            + "<llamadasperdidas>"+Receiver_BootCompleted.var.llamadasperdidas +
            "</llamadasperdidas>";
            log= log
        }
    }
}

```

```

        +"<llamadasrecibidas>"+Receiver_BootCompleted.var.llamadasrecibidas +
        "</llamadasrecibidas>";
        log= log
        +"<llamadasrealizadas>"+Receiver_BootCompleted.var.llamadasrealizadas +
        "</llamadasrealizadas>";
        log= log
        +"<mensajesenviados>"+Receiver_BootCompleted.var.mensajesenviados +
        "</mensajesenviados>";
        log= log
        +"<mensajesrecibidos>"+Receiver_BootCompleted.var.mensajesrecibidos +
        "</mensajesrecibidos>";
        OutputStreamWriter fout =new OutputStreamWriter(new FileOutputStream(f));
        fout.write(log);
        fout.flush();
        fout.close();
    }
}

catch(Exception e)
{
    Log.d("AppDolor", "Error creando fichero: " + nombrefichero);
}

}

public static void creaFichero2(String nombrefichero) {
    try
    {
        String log = null;
        String extStore = Environment.getExternalStorageDirectory().toString();
        File file = new File(extStore +carpeta+ nombrefichero + ".xml");
        if(!file.exists())
        {
            File f = new File(extStore +carpeta+ nombrefichero + ".xml");
            f.createNewFile();
            log = "<?xml version=\"1.0\" encoding=\"UTF-8\"?><results>";
            log= log +"<imei>"+Receiver_BootCompleted.config.imei + "</imei>";
            if (Receiver_BootCompleted.demanda)
            {
                log=log + "<demanda>1</demanda>";
            }
            else
            {
                log=log + "<demanda>0</demanda>";
            }

            log= log +"<fechainicio>"+Receiver_BootCompleted.var.fecha_inicio +
            "</fechainicio>";
            log= log +"<fecha>"+Receiver_BootCompleted.var.fecha + "</fecha>";
            OutputStreamWriter fout =new OutputStreamWriter(new FileOutputStream(f));
            fout.write(log);
            fout.flush();
            fout.close();
        }
    }

    catch(Exception e)
    {
        Log.d("AppDolor", "Error creando fichero: " + nombrefichero);
    }

}

public static void creaFicheroInicial(String nombrefichero) {
    try
    {
        String log = null;
        String extStore = Environment.getExternalStorageDirectory().toString();
        File file = new File(extStore +carpeta+ nombrefichero + ".xml");
    }
}

```

```

    if(!file.exists())
    {
        File f = new File(extStore +carpeta+ nombrefichero + ".xml");
        f.createNewFile();
        log = "<?xml version=\"1.0\" encoding=\"UTF-8\"?><results>";
        log= log + "<imei>" +Receiver_BootCompleted.config.imei + "</imei>";
        if (Receiver_BootCompleted.demanda)
        {
            log=log + "<demanda>1</demanda>";
        }
        else
        {
            log=log + "<demanda>0</demanda>";
        }

        log= log + "<fechainicio>" +Receiver_BootCompleted.var.fecha_inicio +
        "</fechainicio>";
        log= log + "<fecha>" +Receiver_BootCompleted.var.fecha + "</fecha>";
        OutputStreamWriter fout =new OutputStreamWriter(new FileOutputStream(f));
        fout.write(log);
        fout.flush();
        fout.close();
    }
}

catch(Exception e)
{
    Log.d("AppDolor", "Error creando fichero: " + nombrefichero);
}
}

public static void escribeRespuestaFichero(String nombrefichero,String preguntaid, int
valor)
{
    String log = null;
    String extStore = Environment.getExternalStorageDirectory().toString();
    File file = new File(extStore +carpeta+ nombrefichero + ".xml");
    try
    {
        FileWriter f=new FileWriter(file,true);
        log= "<pregunta><id>" +preguntaid + "</id>";
        log= log + "<valor>" +valor + "</valor></pregunta>";
        f.write(log);
        f.flush();
        f.close();
    }
    catch (IOException e)
    {
        Log.d("AppContextual", "Error escribiendo fichero: " + nombrefichero);
    }
}

public static void escribeRespuestaFichero(String nombrefichero,String preguntaid,String
respuesta)
{
    String log = null;
    String extStore = Environment.getExternalStorageDirectory().toString();
    File file = new File(extStore +carpeta+ nombrefichero + ".xml");
    try
    {
        FileWriter f=new FileWriter(file,true);
        log= "<pregunta><id>" +preguntaid + "</id>";
        log= log + "<valor>" +respuesta + "</valor></pregunta>";
        f.write(log);
        f.flush();
        f.close();
    }
}

```

```

        catch (IOException e)
        {
            Log.d("AppContextual", "Error escribiendo fichero: " + nombrefichero);
        }
    }

public static void escribeTiempoFichero(String nombrefichero, String wname, String
description, String temp, String humidity, String pressure, String speed) {

    String log = null;
    String extStore = Environment.getExternalStorageDirectory().toString();
    File file = new File(extStore + carpeta + nombrefichero + ".xml");
    try
    {
        FileWriter f=new FileWriter(file, true);
        log= "<clima><ciudad>" + wname + "</ciudad>";
        log= log + "<descripcion>" + description + "</descripcion>";
        log= log + "<temperatura>" + temp + "</temperatura>";
        log= log + "<humedad>" + humidity + "</humedad>";
        log= log + "<presion>" + pressure + "</presion>";
        log= log + "<viento>" + speed + "</viento></clima>";
        f.write(log);
        f.flush();
        f.close();
    }
    catch (IOException e)
    {
        Log.d("AppContextual", "Error escribiendo fichero: " + nombrefichero);
    }
}

public static void cerrarFicheroResultados(String nombrefichero)
{
    String log = null;
    String extStore = Environment.getExternalStorageDirectory().toString();
    File file = new File(extStore + carpeta + nombrefichero + ".xml");
    try
    {
        FileWriter f=new FileWriter(file, true);
        log= "</results>";
        f.write(log);
        f.flush();
        f.close();
    }
    catch (IOException e)
    {
        Log.d("AppDolor", "Error escribiendo fichero: " + nombrefichero);
    }
}

public static void borrarFichero(String nombrefichero)
{
    String extStore = Environment.getExternalStorageDirectory().toString();
    File file = new File(extStore + carpeta + nombrefichero + ".xml");
    if(file.exists())
    {
        file.delete();
    }
}

public static String ponerAlarma(Context ctx) {

    Calendar fecha = Calendar.getInstance();
    Calendar hoy = Calendar.getInstance();
    Date fechalanzamiento=new Date();
    if (Receiver_BootCompleted.config.contador ==-1)
    {

```

```

Receiver_BootCompleted.config.contador=0;
fechalanzamiento=Receiver_BootCompleted.config .lanzamientos.get(0).hora;
fecha.setTime(fechalanzamiento );
fecha.set(hoy.get(Calendar.YEAR), hoy.get(Calendar.MONTH),
hoy.get(Calendar.DAY_OF_MONTH) );
fecha.set(Calendar.DAY_OF_YEAR, fecha.get(Calendar.DAY_OF_YEAR) + 1);
}
else
{
fechalanzamiento=Receiver_BootCompleted.config
.lanzamientos.get(Receiver_BootCompleted.config.contador).hora;
fecha.setTime(fechalanzamiento );
fecha.set(hoy.get(Calendar.YEAR), hoy.get(Calendar.MONTH),
hoy.get(Calendar.DAY_OF_MONTH) );
}

```

```

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd_HH-mm-ss");
String fechastring = sdf.format(fecha.getTime());
Log.d("AppContextual", "Fecha de alarma: " + fechastring);
AlarmManager alarmManager = (AlarmManager)
ctx.getSystemService(Context.ALARM_SERVICE);
Intent intent = new Intent(ctx, RecibeAlarma.class);
PendingIntent pendingIntent1 = PendingIntent.getBroadcast(ctx, 1,intent,
PendingIntent.FLAG_CANCEL_CURRENT);
alarmManager.set(AlarmManager.RTC_WAKEUP, fecha.getTime().getTime(),pendingIntent1);
return fechastring;
}

```

```

public static String ponerAlarmaDiaSiguiente(Context ctx) {

```

```

Calendar fecha = Calendar.getInstance();
Calendar hoy = Calendar.getInstance();

```

```

fecha.set(Calendar.HOUR_OF_DAY, 0);
fecha.set(Calendar.MINUTE , 0);
fecha.set(Calendar.SECOND, 0);
fecha.set(hoy.get(Calendar.YEAR), hoy.get(Calendar.MONTH),
hoy.get(Calendar.DAY_OF_MONTH) );
fecha.set(Calendar.DAY_OF_YEAR, fecha.get(Calendar.DAY_OF_YEAR) + 1);

```

```

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd_HH-mm-ss");
String fechastring = sdf.format(fecha.getTime());
Log.d("AppContextual", "Proxima vez comprobar fechas: " + fechastring);
AlarmManager alarmManager = (AlarmManager)
ctx.getSystemService(Context.ALARM_SERVICE);
Intent intent = new Intent(ctx, RecibeAlarma2.class);
PendingIntent pendingIntent1 = PendingIntent.getBroadcast(ctx, 1,intent,
PendingIntent.FLAG_CANCEL_CURRENT);
alarmManager.set(AlarmManager.RTC_WAKEUP, fecha.getTime().getTime(),pendingIntent1);
return fechastring;
}

```

```

public static void ponerProximaAlarma(Context ctx, int cont) {

```

```

Calendar fecha = Calendar.getInstance();
Calendar hoy = Calendar.getInstance();
Date fechalanzamiento=new Date();

```

```

if (cont==Receiver_BootCompleted.config.lanzamientos.size())
{

```

```

fechalanzamiento=Receiver_BootCompleted.config .lanzamientos.get(0).hora;
fecha.setTime(fechalanzamiento );
fecha.set(hoy.get(Calendar.YEAR), hoy.get(Calendar.MONTH),
hoy.get(Calendar.DAY_OF_MONTH) );
fecha.set(Calendar.DAY_OF_YEAR, fecha.get(Calendar.DAY_OF_YEAR) +1);
}

```

```

else
{

```

```

fechalanzamiento=Receiver_BootCompleted.config .lanzamientos.get(cont).hora;
fecha.setTime(fechalanzamiento );

```

```

        fecha.set(hoy.get(Calendar.YEAR), hoy.get(Calendar.MONTH),
        hoy.get(Calendar.DAY_OF_MONTH) );
    }

    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd_HH-mm-ss");
    String fechastring = sdf.format(fecha.getTime());
    Log.d("AppContextual", "Fecha de proxima alarma: " + fechastring);
    AlarmManager alarmManager = (AlarmManager) ctx.getSystemService(Context.ALARM_SERVICE);
    Intent intent = new Intent(ctx, RecibeAlarma.class);
    PendingIntent pendingIntent1 = PendingIntent.getBroadcast(ctx, 1, intent,
    PendingIntent.FLAG_CANCEL_CURRENT);
    alarmManager.set(AlarmManager.RTC_WAKEUP, fecha.getTime().getTime(), pendingIntent1);
}

public static void comprobartiempo(final Context context) {

    // Mirar el tiempo si hay conexion
    if (Utils.isOnline(context))
    {

        // obtener valores
        boolean providerEnable = false;
        tiempocalculado=false;
        try {

            // Get a LocationManager
            manager = (LocationManager)
            context.getSystemService(Context.LOCATION_SERVICE);
            listener = new MyLocationListener(context);

            // Inicializamos los servicios
            // If the GPS_PROVIDER is enable use it
            if (manager.isProviderEnabled(LocationManager.GPS_PROVIDER))
            {

                manager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 100, 0,
                listener);
                Log.d("AppDolor", LocationManager.GPS_PROVIDER);
                providerEnable = true;

            }

            // if the NETWORK_PROVIDER is enabled use it
            if (manager.isProviderEnabled(LocationManager.NETWORK_PROVIDER))
            {

                manager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
                100, 0, listener);
                Log.d("AppDolor", LocationManager.NETWORK_PROVIDER);
                providerEnable = true;

            }

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

        if (providerEnable) {

            //poner contador 5 minutos
            Log.d("AppDolor", "Esperar 4 minutos");
            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                public void run() {
                    if (!tiempocalculado)
                    {
                        terminaraplicacion(context);
                    }
                }
            }, MILISEGUNDOS_ESPERA);
        }
    }
}

```

```

        } else {
            Log.d("AppDolor", "No disponemos del servicio de geolocalizaci-n");
            ObtenerInfoTiempo t = new ObtenerInfoTiempo();
            t.setContext(context);
            t.execute( latitude, longitude);
        }

    }

    else
    {
        //no hay red deo cerrado el fichero de datos m3vil y ya se subir3 cuando
        pueda
        Utils.cerrarFicheroResultados( Receiver_BootCompleted.var .nombrefichero );
        // Utils.borrarFichero("datos");
    }
}

```

```

public static void terminaraplicacion(Context context) {

```

```

    Log.d("AppDolor", "Termino porque han pasado 4 minutos y no tengo coordenadas");

```

```

    manager.removeUpdates(listener);
    ObtenerInfoTiempo t = new ObtenerInfoTiempo();
    t.setContext(context);
    t.execute( latitude, longitude);
}

```

```

public static boolean getGPSStatus(Context ctx){

```

```

    String allowedLocationProviders
    =Settings.System.getString(ctx.getContentResolver(),Settings.System.LOCATION_PROVIDERS_ALL
    OWED);

```

```

    if (allowedLocationProviders == null) {
        allowedLocationProviders = "";
    }

```

```

    if (allowedLocationProviders.contains(LocationManager.NETWORK_PROVIDER))
    {
        return true;
    }

```

```

    if (allowedLocationProviders.contains(LocationManager.GPS_PROVIDER))
    {
        return true;
    }

```

```

    return false;
}

```

```

public static void cancelarNotificacion(Context ctx)

```

```

{
    NotificationManager notificationManager =
    (NotificationManager)ctx.getSystemService(Context.NOTIFICATION_SERVICE);
    notificationManager.cancel(Utils.APP_ID_NOTIFICATION);
}

```

```

public static Drawable loadImageFromWebOperations(String url) {

```

```

    try {
        InputStream is = (InputStream) new URL(url).getContent();
        Drawable d = Drawable.createFromStream(is, "src name");
        return d;
    } catch (Exception e) {
        return null;
    }
}

```

```

}

```

```

public static Drawable loadImageFromFile(String nombreimagen) {
    try {
        String extStore = Environment.getExternalStorageDirectory().toString();
        Drawable d = Drawable.createFromPath(extStore +carpeta+ nombreimagen );
        return d;
    } catch (Exception e) {
        return null;
    }
}

public static Bitmap loadBitmapFromFile(String nombreimagen) {
    try {
        String extStore = Environment.getExternalStorageDirectory().toString();
        Bitmap b = BitmapFactory.decodeFile(extStore +carpeta+ nombreimagen );
        return b;
    } catch (Exception e) {
        return null;
    }
}

public static Bitmap loadBitmapFromFileWithDimensions(String nombreimagen,int height) {
    try {
        String extStore = Environment.getExternalStorageDirectory().toString();
        Bitmap b = BitmapFactory.decodeFile(extStore +carpeta+ nombreimagen );
        float aspectRatio=b.getWidth()/((float) b.getHeight());
        int width=Math.round(height* aspectRatio);
        Bitmap b_scaled=Bitmap.createScaledBitmap(b, width,height,false);
        return b_scaled;
    } catch (Exception e) {
        return null;
    }
}

public static void saveImageFromWebOperations(String url, String nombreimagen) {
    try {
        InputStream is = (InputStream) new URL(url).getContent();
        // Drawable d = Drawable.createFromStream(is, "src name");
        Bitmap image_saved=BitmapFactory.decodeStream(is);

        String extStore = Environment.getExternalStorageDirectory().toString();
        FileOutputStream fOut=new FileOutputStream(extStore +carpeta+ nombreimagen );
        image_saved.compress(Bitmap.CompressFormat.JPEG,100,fOut);
        fOut.flush();
        fOut.close();

    } catch (Exception e) {
    }
}

public static void descargarimagenesfichero(String nombrefichero) {
    String extStore = Environment.getExternalStorageDirectory().toString();
    File file = new File(extStore +carpeta+ nombrefichero + ".xml");
    if(file.exists())
    {
        try {
            //leer campos imagen
            XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
            XmlPullParser xpp = factory.newPullParser();
            FileInputStream fis = new FileInputStream(file);
            xpp.setInput(new InputStreamReader(fis));
            int event = xpp.getEventType();
            while (event != XmlPullParser.END_DOCUMENT) {
                String name=xpp.getName();
                switch (event){

```

```
case
XmlPullParser.START_TAG:
    if (name.equals("imagenrespuesta"))
    {
        String imagen=xpp.nextText();
        File fileimagen=new File(extStore +carpeta+ imagen);
        if (!fileimagen.exists())
        {
            SaveImageFromWebOperations(direccionImagenes + imagen, imagen);
        }
    }
    break;
}
```

```
event = xpp.next();
```

```
}
```

```
} catch (Exception ex)
```

```
{
```

```
ex.printStackTrace();
```

```
}
```

```
}
```

```
}
```

```
}
```

```
package appcontextualv3.code;

public class Variables {

    public int llamadasrealizadas=0;
    public int llamadasrecibidas=0;
    public int llamadasperdidas=0;
    public int mensajesrecibidos=0;
    public int mensajesenviados=0;
    public String fecha;
    public String fecha_inicio;
    public String nombrefichero;

    public void limpiarVariables ()
    {

        llamadasperdidas=0;
        llamadasrealizadas=0;
        llamadasrecibidas=0;
        mensajesenviados=0;
        mensajesrecibidos=0;
        fecha=null;
        fecha_inicio=null;

    }

    public void limpiarVariablesMovil ()
    {

        llamadasperdidas=0;
        llamadasrealizadas=0;
        llamadasrecibidas=0;
        mensajesenviados=0;
        mensajesrecibidos=0;

    }

}
```

```
package appcontextualv3.code;

import java.lang.reflect.Field;
import java.util.ArrayList;
import java.util.List;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.text.SpannableString;
import android.text.Spanned;
import android.text.style.ImageSpan;
import android.util.DisplayMetrics;
import android.util.Log;
import android.util.TypedValue;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewConfiguration;
import android.view.WindowManager;
import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.ScrollView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;

public class VisorPregunta extends AppCompatActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // requestWindowFeature(Window.FEATURE_NO_TITLE);

        // Activity a pantalla completa

        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.LayoutParams.FLAG_FULLSCREEN);
        //barra

        ActionBar actionBar = getSupportActionBar();
        actionBar.setDisplayHomeAsUpEnabled(false);
        // actionBar.setDisplayHomeAsUpEnabled(false);
        actionBar.setBackgroundDrawable(new ColorDrawable(Color.parseColor("#333333")));

        setContentView(R.layout.fondo);
        LinearLayout fondo=(LinearLayout) findViewById(R.id.layoutfondo);

        Log.d("AppDolor", "Demanda: "+Receiver_BootCompleted.demanda );

        final Lanzamiento l;
        if (Receiver_BootCompleted.config.lanzamientos.size()==0 && Receiver_BootCompleted.config.demanda==null)
```

```

{
    Utils.parsearchivoconfiguracion("conf",this.getApplicationContext());
    Utils.parsearchivo("datos");
}

if (Receiver_BootCompleted.demanda)
{
    l=Receiver_BootCompleted.config.demanda;
}
else
{
    SharedPreferences settings =
    getApplicationContext().getSharedPreferences(Utils.appname, 0);
    SharedPreferences.Editor editor = settings.edit();
    editor.putBoolean("notificacion", false);
    editor.putInt("contador", 0);
    editor.commit();
    int cont=getIntent().getExtras().getInt("cont",0);
    Receiver_BootCompleted.config.contador=cont;
    l=Receiver_BootCompleted.config.lanzamientos.get(cont);
}

if (l==null)
{
    Toast.makeText(this, "No tienes habilitada la opción de contestar a demanda",
    Toast.LENGTH_SHORT).show();
    Receiver_BootCompleted.demanda=false;
    Intent intent = new Intent(VisorPregunta.this,Instrucciones.class);
    intent.putExtra("origen",3);
    VisorPregunta.this.startActivity(intent);
    VisorPregunta.this.finish();
}
else
{
    //Utils.parsearchivo("datos");
    boolean b= Utils.existefichero("results2"+Receiver_BootCompleted.var.fecha);
    if (!b)
    {
        Utils.creaFichero2("results2"+Receiver_BootCompleted.var.fecha);
    }

    int pregunta=l.contadorpregunta;
    Pregunta p=l.preguntas.get(pregunta);
    if(p.tipo.equals("Radio"))
    {
        TextView tv=new TextView(this);
        tv.setText(p.texto);
        tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,22);
        tv.setTextColor(Color.parseColor("#676767"));
        fondo.addView(tv);

        TextView tv2=new TextView(this);
        tv2.setText("SELECCIONE UNA OPCIÓN");
        tv2.setTextSize(TypedValue.COMPLEX_UNIT_DIP,14);
        tv2.setTextColor(Color.parseColor("#a1a9ab"));
        LinearLayout.LayoutParams params= new
        LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT ,
        LinearLayout.LayoutParams.WRAP_CONTENT);
        params.setMargins(0,10,0,20);
        tv2.setLayoutParams(params);
        fondo.addView(tv2);
    }
}
}

```

```

ScrollView sv= new ScrollView(this);
sv.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT));
sv.setScrollbarFadingEnabled(false);

LinearLayout ll= new LinearLayout(this);
ll.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.MATCH_PARENT));
ll.setPadding(5, 5, 5, 5);
ll.setOrientation(LinearLayout.VERTICAL );

final ToggleButton[] botones= new ToggleButton[p.numrespuestas];
DisplayMetrics metrics =
getApplicationContext().getResources().getDisplayMetrics();
int width=metrics.widthPixels;
float aspectRatio=800/(float)118;
int altoboton=Math.round(width/aspectRatio);

for (int i=0; i<p.respuestas.size(); i++)
{

    botones[i] = new ToggleButton(this);

    if (p.tiporespuesta==1)
    {
        botones[i].setText(p.respuestas .get(i).texto);
        botones[i].setTextOff(p.respuestas .get(i).texto);
        botones[i].setTextOn(p.respuestas .get(i).texto);
    }

    if (p.tiporespuesta ==2)
    {
        ImageSpan imageSpan = new
        ImageSpan(this,Utills.LoadBitmapFromFileWithDimensions(p.respuestas
        .get(i).imagen ,altoboton));
        SpannableString content = new SpannableString("X");
        content.setSpan(imageSpan, 0,1,Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        botones[i].setText(content);
        botones[i].setTextOff(content);
        botones[i].setTextOn(content);
    }

    if (p.tiporespuesta ==3)
    {
        ImageSpan imageSpan = new
        ImageSpan(this,Utills.LoadBitmapFromFileWithDimensions(p.respuestas
        .get(i).imagen ,altoboton));
        String texto=p.respuestas .get(i).texto;
        SpannableString content = new SpannableString("X "+texto);
        content.setSpan(imageSpan, 0,1,Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        botones[i].setText(content);
        botones[i].setTextOff(content);
        botones[i].setTextOn(content);
    }

    botones[i].setBackgroundResource(R.drawable.botones_style );
    botones[i].setTextSize(TypedValue.COMPLEX_UNIT_DIP,18);
    botones[i].setTextColor(Color.parseColor("#676767"));
    botones[i].setId(i);
    botones[i].setGravity(Gravity.LEFT|Gravity.CENTER);
    botones[i].setPadding(20, 0, 75,0);
    ll.addView(botones[i]);
    botones[i].setOnCheckedChangeListener(new
    CompoundButton.OnCheckedChangeListener() {

        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
        isChecked) {
            if (isChecked)
            {

```

```

        int indice=buttonView.getId();
        botones[indice].setTextColor(Color.parseColor("#2C88BF"));
        for(int j=0; j<botones.length; j++)
        {
            if (j!=indice)
            {
                botones[j].setChecked(false);

                botones[j].setTextColor(Color.parseColor("#676767"));
            }
        }
    }
    else
    {
        int indice=buttonView.getId();
        botones[indice].setTextColor(Color.parseColor("#676767"));
    }
}
});
}

```

```

Button salir= new Button(this);
salir.setText("OK");
salir.setTextSize(TypedValue.COMPLEX_UNIT_DIP,30);
salir.setBackgroundResource(R.drawable.botonazul);
salir.setTextColor(Color.WHITE);
LinearLayout.LayoutParams params2= new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT ,
LinearLayout.LayoutParams.WRAP_CONTENT);
params2.setMargins(0,50,0,0);
salir.setLayoutParams(params2);
ll.addView(salir);

```

```

salir.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        String preguntaid=l.preguntas.get(l.contadorpregunta).id;
        int marcado=-1;
        for (int i=0; i<botones.length; i++)
        {
            if (botones[i].isChecked())
            {
                marcado=i;
            }
        }

        if (marcado!=-1)
        {
            int valor=l.preguntas.get(l.contadorpregunta).respuestas
            .get(marcado).valor;
            Log.d("AppDolor", "valor seleccionado: "+valor);

            Utils.escribeRespuestaFichero("results2"+Receiver_BootComp
            leted.var.fecha,preguntaid,valor);
            l.contadorpregunta=l.contadorpregunta+1;
            if (l.contadorpregunta<l.preguntas.size())
            {
                //siguiente pregunta
                Intent intent=new
                Intent(VisorPregunta.this,VisorPregunta.class);

                intent.putExtra("cont",Receiver_BootCompleted.config.c
                ontador );
                VisorPregunta.this.startActivity(intent);
                VisorPregunta.this.finish();
            }
        }
    }
}

```

```

        else
        {
            //fin preguntas.
            l.contadorpregunta=0;
            VisorPregunta.this.startActivity(new
            Intent(VisorPregunta.this,FinPreguntas.class));
            VisorPregunta.this.finish();
        }
    }
});

sv.addView(ll);

fondo.addView(sv);
}

if(p.tipo.equals("Texto"))
{

RelativeLayout rl= new RelativeLayout(this);
rl.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.MATCH_PARENT));
rl.setPadding(5, 5, 5, 5);

TextView tv=new TextView(this);
tv.setText(p.texto);
tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,22);
tv.setTextColor(Color.parseColor("#676767"));
rl.addView(tv);

final EditText et= new EditText(this);
et.setSingleLine(false);
et.setGravity(Gravity.TOP|Gravity.LEFT);
et.setBackgroundResource(R.drawable.cuadrotextog);
et.setVerticalScrollBarEnabled(true);
et.setTextColor(Color.parseColor("#b6b6b6"));
RelativeLayout.LayoutParams params= new
RelativeLayout.LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT);
params.addRule(RelativeLayout.CENTER_VERTICAL);
et.setLayoutParams(params);
rl.addView(et);

Button bt = new Button(this);
bt.setText("OK");
bt.setTextSize(TypedValue.COMPLEX_UNIT_DIP,30);
bt.setBackgroundResource(R.drawable.botonazul);
bt.setTextColor(Color.WHITE);
RelativeLayout.LayoutParams params2= new
RelativeLayout.LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT);
params2.setMargins(0,40,0,10);
params2.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM);
bt.setLayoutParams(params2);
rl.addView(bt);
bt.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View v) {

String preguntaid=l.preguntas.get(l.contadorpregunta).id;
String valor=et.getText().toString();
Log.d("AppDolor", "texto escrito: "+valor);

Utils.escribeRespuestaFichero("results2"+Receiver_BootCompleted.var.fe
cha,preguntaid,valor);

```

```

        l.contadorpregunta=l.contadorpregunta+1;
        if (l.contadorpregunta<l.preguntas.size())
        {
            //siguiente pregunta
            Intent intent=new Intent(VisorPregunta.this,VisorPregunta.class);
            intent.putExtra("cont",Receiver_BootCompleted.config.contador );
            VisorPregunta.this.startActivity(intent);
            VisorPregunta.this.finish();
        }
        else
        {
            //fin preguntas.
            l.contadorpregunta=0;
            VisorPregunta.this.startActivity(new
            Intent(VisorPregunta.this,FinPreguntas.class));
            VisorPregunta.this.finish();
        }
    }
});

fondo.addView(rl);

}

if(p.tipo.equals("Slider"))
{

    final RelativeLayout rl= new RelativeLayout(this);
    rl.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
    LayoutParams.MATCH_PARENT));
    rl.setPadding(5, 5, 5, 5);

    TextView tv=new TextView(this);
    tv.setId(1);
    tv.setText(p.texto);
    tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,22);
    tv.setTextColor(Color.parseColor("#676767"));
    rl.addView(tv);

    final TextView tv3=new TextView(getApplicationContext());
    tv3.setId(5);
    tv3.setText("");
    tv3.setTextSize(TypedValue.COMPLEX_UNIT_DIP,16);
    tv3.setTextColor(Color.parseColor("#2C88BF"));
    RelativeLayout.LayoutParams paramstv3= new
    RelativeLayout.LayoutParams(LayoutParams.WRAP_CONTENT ,
    LayoutParams.WRAP_CONTENT);
    paramstv3.setMargins(0,50,0,10);
    paramstv3.addRule(RelativeLayout.BELOW,tv.getId());
    paramstv3.addRule(RelativeLayout.CENTER_HORIZONTAL);
    tv3.setLayoutParams( paramstv3);
    rl.addView(tv3);

    final SeekBar sb= new SeekBar(this);
    sb.setId(2);
    sb.setMax(p.numrespuestas-1);
    sb.setProgress(p.numrespuestas /2);
    sb.setProgressDrawable(getResources().getDrawable(R.drawable .progress_bar));

    if (p.tiporespuesta ==1)
    {
        final TextView tv1=new TextView(this);

```

```

tv1.setId(3);
tv1.setText(p.respuestas.get(0).texto);
tv1.setTextSize(TypedValue.COMPLEX_UNIT_DIP,14);
tv1.setTextColor(Color.parseColor("#676767"));
RelativeLayout.LayoutParams paramstv1= new
RelativeLayout.LayoutParams (LayoutParams.WRAP_CONTENT ,
LayoutParams.WRAP_CONTENT);
paramstv1.setMargins(0,40,0,10);
paramstv1.addRule(RelativeLayout.BELOW,tv3.getId());
paramstv1.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
tv1.setLayoutParams ( paramstv1);
rl.addView(tv1);

TextView tv2=new TextView(this);
tv2.setId(4);
tv2.setText(p.respuestas.get(p.numrespuestas-1).texto);
tv2.setTextSize(TypedValue.COMPLEX_UNIT_DIP,14);
tv2.setTextColor(Color.parseColor("#676767"));
RelativeLayout.LayoutParams paramstv2= new
RelativeLayout.LayoutParams (LayoutParams.WRAP_CONTENT ,
LayoutParams.WRAP_CONTENT);
paramstv2.setMargins(0,40,0,10);
paramstv2.addRule(RelativeLayout.BELOW,tv3.getId());
paramstv2.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
tv2.setLayoutParams ( paramstv2);
rl.addView(tv2);

RelativeLayout.LayoutParams params= new
RelativeLayout.LayoutParams (LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT);
params.setMargins(0,10,0,10);
params.addRule(RelativeLayout.BELOW,tv1.getId());
sb.setLayoutParams (params);

}

if (p.tiporespuesta ==2)
{
    DisplayMetrics metrics =
    getApplicationContext().getResources().getDisplayMetrics();
    int width = metrics.widthPixels;
    int maximo= width/3;
    int tamaño=maximo - 150;

    if (tamaño>250)
    {
        tamaño=250;
    }

    if (tamaño<50)
    {
        tamaño=50;
    }

    final ImageView iv1=new ImageView(this);
    iv1.setId(6);
    iv1.setImageDrawable (Utils.LoadImageFromFile ( p.respuestas
    .get(0).imagen ));
    RelativeLayout.LayoutParams paramsiv1= new
    RelativeLayout.LayoutParams (tamaño , tamaño);
    paramsiv1.setMargins(0,40,0,10);
    paramsiv1.addRule(RelativeLayout.BELOW,tv3.getId());
    paramsiv1.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
    iv1.setLayoutParams ( paramsiv1);
    rl.addView(iv1);

    ImageView iv2=new ImageView(this);
    iv2.setId(7);
    iv2.setImageDrawable (Utils.LoadImageFromFile ( p.respuestas
    .get(p.numrespuestas /2).imagen ));

```

```

RelativeLayout.LayoutParams paramsiv2= new
RelativeLayout.LayoutParams(tamanyo , tamanyo);
paramsiv2.setMargins(0,40,0,10);
paramsiv2.addRule(RelativeLayout.BELOW,tv3.getId());
paramsiv2.addRule(RelativeLayout.CENTER_HORIZONTAL );
iv2.setLayoutParams ( paramsiv2);
rl.addView(iv2);

ImageView iv3=new ImageView(this);
iv3.setId(8);
iv3.setImageDrawable(Utils.LoadImageFromFile( p.respuestas
.get(p.numrespuestas-1).imagen ));
RelativeLayout.LayoutParams paramsiv3= new
RelativeLayout.LayoutParams(tamanyo , tamanyo);
paramsiv3.setMargins(0,40,0,10);
paramsiv3.addRule(RelativeLayout.BELOW,tv3.getId());
paramsiv3.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
iv3.setLayoutParams ( paramsiv3);
rl.addView(iv3);

RelativeLayout.LayoutParams params= new
RelativeLayout.LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT);
params.setMargins(0,10,0,10);
params.addRule(RelativeLayout.BELOW,iv2.getId());
sb.setLayoutParams(params);
}
rl.addView(sb);

sb.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {

@Override
public void onStopTrackingTouch(SeekBar seekBar) {
// TODO Auto-generated method stub
}

@Override
public void onStartTrackingTouch(SeekBar seekBar) {
// TODO Auto-generated method stub
}

@Override
public void onProgressChanged(SeekBar seekBar, int progress,boolean
fromUser) {
// TODO Auto-generated method stub

rl.removeView(tv3);
int
valor=l.preguntas.get(l.contadorpregunta).respuestas.get(progress).val
or;
tv3.setText("Valor seleccionado: "+ valor);
rl.addView(tv3);
}
});

Button bt = new Button(this);
bt.setText("OK");
bt.setTextSize(TypedValue.COMPLEX_UNIT_DIP,30);
bt.setBackgroundResource(R.drawable.botonazul);
bt.setTextColor(Color.WHITE);
RelativeLayout.LayoutParams params2= new
RelativeLayout.LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT);
params2.setMargins(0,40,0,10);
params2.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM);

```

```

bt.setLayoutParams(params2);
rl.addView(bt);
bt.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        String preguntaid=l.preguntas.get(l.contadorpregunta).id;
        int valorsb=sb.getProgress();
        int
        valor=l.preguntas.get(l.contadorpregunta).respuestas.get(valorsb).valor;
        Log.d("AppDolor", "slider seleccionado: "+valor);

        Utils.escribeRespuestaFichero("results2"+Receiver_BootCompleted.var.fecha,preguntaid,valor);
        l.contadorpregunta=l.contadorpregunta+1;
        if (l.contadorpregunta<l.preguntas.size())
        {
            //siguiente pregunta
            Intent intent=new Intent(VisorPregunta.this,VisorPregunta.class);
            intent.putExtra("cont",Receiver_BootCompleted.config.contador);
            VisorPregunta.this.startActivity(intent);
            VisorPregunta.this.finish();
        }
        else
        {
            //fin preguntas.
            l.contadorpregunta=0;
            VisorPregunta.this.startActivity(new
            Intent(VisorPregunta.this,FinPreguntas.class));
            VisorPregunta.this.finish();
        }
    }
});

fondo.addView(rl);

}
if(p.tipo.equals("Check"))
{

    TextView tv=new TextView(this);
    tv.setText(p.texto);
    tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,22);
    tv.setTextColor(Color.parseColor("#676767"));
    fondo.addView(tv);

    TextView tv2=new TextView(this);
    tv2.setText("SELECCIONE LAS OPCIONES");
    tv2.setTextSize(TypedValue.COMPLEX_UNIT_DIP,14);
    tv2.setTextColor(Color.parseColor("#a1a9ab"));
    LinearLayout.LayoutParams params= new
    LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT ,
    LinearLayout.LayoutParams.WRAP_CONTENT);
    params.setMargins(0,10,0,20);
    tv2.setLayoutParams(params);
    fondo.addView(tv2);

    ScrollView sv= new ScrollView(this);
    sv.setLayoutParams(new LinearLayout.LayoutParams(LinearLayout.MATCH_PARENT ,
    LinearLayout.LayoutParams.WRAP_CONTENT));
    sv.setScrollbarFadingEnabled(false);

    LinearLayout ll= new LinearLayout(this);
    ll.setLayoutParams(new LinearLayout.LayoutParams(LinearLayout.MATCH_PARENT ,
    LinearLayout.LayoutParams.MATCH_PARENT));

```

```

ll.setPadding(5, 5, 5, 5);
ll.setOrientation(LinearLayout.VERTICAL );

final ToggleButton[] botones= new ToggleButton[p.numrespuestas];
DisplayMetrics metrics =
getApplicationContext().getResources().getDisplayMetrics();
int width=metrics.widthPixels;
float aspectRatio=800/(float)118;
int altoboton=Math.round(width/aspectRatio);

for (int i=0; i<p.respuestas.size(); i++)
{

    botones[i] = new ToggleButton(this);

    if (p.tiporespuesta==1)
    {
        botones[i].setText(p.respuestas .get(i).texto);
        botones[i].setTextOff(p.respuestas .get(i).texto);
        botones[i].setTextOn(p.respuestas .get(i).texto);
    }

    if (p.tiporespuesta ==2)
    {
        ImageSpan imageSpan = new
        ImageSpan(this,Utills.LoadBitmapFromFileWithDimensions(p.respuestas
        .get(i).imagen ,altoboton));
        SpannableString content = new SpannableString("X");
        content.setSpan(imageSpan, 0,1,Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        botones[i].setText(content);
        botones[i].setTextOff(content);
        botones[i].setTextOn(content);
    }

    if (p.tiporespuesta ==3)
    {
        ImageSpan imageSpan = new
        ImageSpan(this,Utills.LoadBitmapFromFileWithDimensions(p.respuestas
        .get(i).imagen ,altoboton));
        String texto=p.respuestas .get(i).texto;
        SpannableString content = new SpannableString("X "+texto);
        content.setSpan(imageSpan, 0,1,Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        botones[i].setText(content);
        botones[i].setTextOff(content);
        botones[i].setTextOn(content);
    }

    botones[i].setBackgroundResource(R.drawable.botones_style );
    botones[i].setTextSize(TypedValue.COMPLEX_UNIT_DIP,18);
    botones[i].setTextColor(Color.parseColor("#676767"));
    botones[i].setId(i);
    botones[i].setGravity(Gravity.LEFT|Gravity.CENTER);
    botones[i].setPadding(20, 0, 75,0);
    ll.addView(botones[i]);
    botones[i].setOnCheckedChangeListener(new
    CompoundButton.OnCheckedChangeListener() {

        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
        isChecked) {
            if (isChecked)
            {
                int indice=buttonView.getId();
                botones[indice].setTextColor(Color.parseColor("#2C88BF"));
            }
            else
            {
                int indice=buttonView.getId();
                botones[indice].setTextColor(Color.parseColor("#676767"));
            }
        }
    }
}

```

```

    });
}

Button salir= new Button(this);
salir.setText("OK");
salir.setTextSize(TypedValue.COMPLEX_UNIT_DIP,30);
salir.setBackgroundResource(R.drawable.botonazul);
salir.setTextColor(Color.WHITE);
LinearLayout.LayoutParams params2= new
LinearLayout.LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT);
params2.setMargins(0,50,0,0);
salir.setLayoutParams(params2);
ll.addView(salir);

salir.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        String preguntaid=l.preguntas.get(l.contadorpregunta).id;
        List<Integer> marcados=new ArrayList<Integer>();
        for (int i=0; i<botones.length; i++)
        {
            if (botones[i].isChecked())
            {
                marcados.add(i);
            }

        }

        if (marcados.size(>0))
        {
            String valor="";
            for (int marcado:marcados)
            {
                if (valor.isEmpty())
                {
                    valor=Integer.toString(l.preguntas.get(l.contadorpr
egunta).respuestas .get(marcado).valor);
                }
                else
                {
                    valor=valor+"*"+Integer.toString(
l.preguntas.get(l.contadorpregunta).respuestas
.get(marcado).valor);
                }
            }

            Log.d("AppPruebaUsabilidad", "valor seleccionado:
"+valor);

            Utils.escribeRespuestaFichero("results2"+Receiver_BootComp
leted.var.fecha,preguntaid,valor);
            l.contadorpregunta=l.contadorpregunta+1;
            if (l.contadorpregunta<l.preguntas.size())
            {
                //siguiente pregunta
                Intent intent=new
                Intent(VisorPregunta.this,VisorPregunta.class);

                intent.putExtra("cont",Receiver_BootCompleted.config.c
ontador );
                VisorPregunta.this.startActivity(intent);
                VisorPregunta.this.finish();
            }
            else
            {

```

```

        //fin preguntas.
        l.contadorpregunta=0;
        VisorPregunta.this.startActivity(new
        Intent(VisorPregunta.this,FinPreguntas.class));
        VisorPregunta.this.finish();
    }
}

});

sv.addView(ll);

fondo.addView(sv);
}
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.preguntas, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected (MenuItem item) {
    switch(item.getItemId()){
        case R.id.action_settings :
            Intent intent=new Intent(VisorPregunta.this,CambioHora.class);
            intent.putExtra("origen",2 );
            VisorPregunta.this.startActivity(intent);
            VisorPregunta.this.finish();
            return true;

        case R.id.action_instructions :
            Intent intent2=new Intent(VisorPregunta.this,Instrucciones.class);
            intent2.putExtra("origen",2 );
            VisorPregunta.this.startActivity(intent2);
            VisorPregunta.this.finish();
            return true;
    }
    return super.onOptionsItemSelected( item);
}
}
}

```

```
package appcontextualv3.code;

import java.util.ArrayList;
import java.util.List;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.StrictMode;
import android.support.v7.app.ActionBar;
import android.text.SpannableString;
import android.text.Spanned;
import android.text.style.ImageSpan;
import android.util.DisplayMetrics;
import android.util.Log;
import android.util.TypedValue;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.ScrollView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;

public class VisorPreguntaInicial extends Activity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);

        // Activity a pantalla completa

        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.Lay
outParams.FLAG_FULLSCREEN);

        setContentView(R.layout.fondo);
        LinearLayout fondo=(LinearLayout) findViewById(R.id.layoutfondo);

        final Lanzamiento l=Utils.parsearficheroconfiguracioninicial("confinicial",
getApplicationContext());

        if (l==null)
        {
            SharedPreferences settings =
getApplicationContext().getSharedPreferences(Utils.appname, 0);
            SharedPreferences.Editor editor = settings.edit();
            editor.putBoolean("inicial", true);
            editor.commit();
            Intent intent=new Intent(VisorPreguntaInicial.this,CambioHora.class);
            intent.putExtra("origen",1 );
            VisorPreguntaInicial .this.startActivity(intent);
            VisorPreguntaInicial.this.finish();
        }
        else
        {
            if (l.preguntas .size()==0)
            {
```

```

    SharedPreferences settings =
    getApplicationContext().getSharedPreferences(Utils.appname, 0);
    SharedPreferences.Editor editor = settings.edit();
    editor.putBoolean("inicial", true);
    editor.commit();
    Intent intent=new Intent(VisorPreguntaInicial.this,CambioHora.class);
    intent.putExtra("origen",1 );
    VisorPreguntaInicial .this.startActivity(intent);
    VisorPreguntaInicial.this.finish();
}

else
{

int cont=getIntent().getExtras().getInt("cont",0);
l.contadorpregunta =cont;
boolean b= Utils.existefichero("resultsinicial");
if (!b)
{
    Utils.creaFicheroInicial("resultsinicial");
}

int pregunta=l.contadorpregunta;
Pregunta p=l.preguntas.get(pregunta);
if(p.tipo.equals("Radio"))
{

    TextView tv=new TextView(this);
    tv.setText(p.texto);
    tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,22);
    tv.setTextColor(Color.parseColor("#676767"));
    fondo.addView(tv);

    TextView tv2=new TextView(this);
    tv2.setText("SELECCIONE UNA OPCIÓN");
    tv2.setTextSize(TypedValue.COMPLEX_UNIT_DIP,14);
    tv2.setTextColor(Color.parseColor("#a1a9ab"));
    LinearLayout.LayoutParams params= new
    LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT ,
    LinearLayout.LayoutParams.WRAP_CONTENT);
    params.setMargins(0,10,0,20);
    tv2.setLayoutParams(params);
    fondo.addView(tv2);

    ScrollView sv= new ScrollView(this);
    sv.setLayoutParams(new LinearLayout.LayoutParams(LinearLayout.MATCH_PARENT ,
    LinearLayout.LayoutParams.WRAP_CONTENT));
    sv.setScrollbarFadingEnabled(false);

    LinearLayout ll= new LinearLayout(this);
    ll.setLayoutParams(new LinearLayout.LayoutParams(LinearLayout.MATCH_PARENT ,
    LinearLayout.LayoutParams.MATCH_PARENT));
    ll.setPadding(5, 5, 5, 5);
    ll.setOrientation(LinearLayout.VERTICAL );

    final ToggleButton[] botones= new ToggleButton[p.numrespuestas];
    DisplayMetrics metrics =
    getApplicationContext().getResources().getDisplayMetrics();
    int width=metrics.widthPixels;
    float aspectRatio=800/(float)118;
    int altoboton=Math.round(width/aspectRatio);
    for (int i=0; i<p.respuestas.size(); i++)
    {

        botones[i] = new ToggleButton(this);
        if (p.tiporespuesta==1)
        {

```

```

        botones[i].setText(p.respuestas .get(i).texto);
        botones[i].setTextOff(p.respuestas .get(i).texto);
        botones[i].setTextOn(p.respuestas .get(i).texto);
    }

    if (p.tiporespuesta ==2)
    {
        ImageSpan imageSpan = new
        ImageSpan(this,Utills.LoadBitmapFromFileWithDimensions (p.respuestas
        .get(i).imagen ,altoboton));
        SpannableString content = new SpannableString("X");
        content.setSpan(imageSpan, 0,1,Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        botones[i].setText(content);
        botones[i].setTextOff(content);
        botones[i].setTextOn(content);
    }

    if (p.tiporespuesta ==3)
    {
        ImageSpan imageSpan = new
        ImageSpan(this,Utills.LoadBitmapFromFileWithDimensions (p.respuestas
        .get(i).imagen ,altoboton));
        String texto=p.respuestas .get(i).texto;
        SpannableString content = new SpannableString("X "+texto);
        content.setSpan(imageSpan, 0,1,Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        botones[i].setText(content);
        botones[i].setTextOff(content);
        botones[i].setTextOn(content);
    }
    botones[i].setBackgroundResource (R.drawable.botones_style );
    botones[i].setTextSize (TypedValue.COMPLEX_UNIT_DIP,18);
    botones[i].setTextColor (Color.parseColor ("#676767"));
    botones[i].setId(i);
    botones[i].setGravity (Gravity.LEFT|Gravity.CENTER);
    botones[i].setPadding(20, 0, 75,0);
    ll.addView(botones[i]);
    botones[i].setOnCheckedChangeListener (new
    CompoundButton.OnCheckedChangeListener () {

        @Override
        public void onCheckedChanged (CompoundButton buttonView, boolean
        isChecked) {
            if (isChecked)
            {
                int indice=buttonView.getId();
                botones[indice].setTextColor (Color.parseColor ("#2C88BF"));
                for(int j=0; j<botones.length; j++)
                {
                    if (j!=indice)
                    {
                        botones[j].setChecked (false);

                        botones[j].setTextColor (Color.parseColor ("#676767"
                        ));
                    }
                }
            }
            else
            {
                int indice=buttonView.getId();
                botones[indice].setTextColor (Color.parseColor ("#676767"));
            }
        }
    });
}

Button salir= new Button(this);
salir.setText ("OK");
salir.setTextSize (TypedValue.COMPLEX_UNIT_DIP,30);
salir.setBackgroundResource (R.drawable.botonazul);

```

```

salir.setTextColor(Color.WHITE);
LinearLayout.LayoutParams params2= new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT ,
LinearLayout.LayoutParams.WRAP_CONTENT);
params2.setMargins(0,50,0,0);
salir.setLayoutParams(params2);
ll.addView(salir);

salir.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        String preguntaid=l.preguntas.get(l.contadorpregunta).id;
        int marcado=-1;
        for (int i=0; i<botones.length; i++)
        {
            if (botones[i].isChecked())
            {
                marcado=i;
            }

        }

        if (marcado!=-1)
        {
            int valor=l.preguntas.get(l.contadorpregunta).respuestas
            .get(marcado).valor;
            Log.d("AppDolor", "valor seleccionado: "+valor);

            Utils.escribeRespuestaFichero("resultsinicial",preguntaid,
            valor);
            l.contadorpregunta=l.contadorpregunta+1;
            if (l.contadorpregunta<l.preguntas.size())
            {
                //siguiente pregunta
                Intent intent=new
                Intent(VisorPreguntaInicial.this,VisorPreguntaInicial.
                class);
                intent.putExtra("cont",l.contadorpregunta );
                VisorPreguntaInicial.this.startActivity(intent);
                VisorPreguntaInicial.this.finish();
            }
            else
            {
                //fin preguntas.
                l.contadorpregunta=0;
                new Wait2().execute("");
            }
        }

    }

});

sv.addView(ll);

fondo.addView(sv);
}

if(p.tipo.equals("Texto"))
{

RelativeLayout rl= new RelativeLayout(this);
rl.setLayoutParams(new LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT ,
LinearLayout.LayoutParams.MATCH_PARENT));
rl.setPadding(5, 5, 5, 5);

TextView tv=new TextView(this);

```

```

tv.setText(p.texto);
tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,22);
tv.setTextColor(Color.parseColor("#676767"));
rl.addView(tv);

final EditText et= new EditText(this);
et.setSingleLine(false);
et.setGravity(Gravity.TOP|Gravity.LEFT);
et.setBackgroundResource(R.drawable.cuadrotextog);
et.setVerticalScrollBarEnabled(true);
et.setTextColor(Color.parseColor("#b6b6b6"));
RelativeLayout.LayoutParams params= new
RelativeLayout.LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT);
params.addRule(RelativeLayout.CENTER_VERTICAL);
et.setLayoutParams(params);
rl.addView(et);

Button bt = new Button(this);
bt.setText("OK");
bt.setTextSize(TypedValue.COMPLEX_UNIT_DIP,30);
bt.setBackgroundResource(R.drawable.botonazul);
bt.setTextColor(Color.WHITE);
RelativeLayout.LayoutParams params2= new
RelativeLayout.LayoutParams(LayoutParams.MATCH_PARENT ,
LayoutParams.WRAP_CONTENT);
params2.setMargins(0,40,0,10);
params2.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM);
bt.setLayoutParams(params2);
rl.addView(bt);
bt.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        String preguntaid=l.preguntas.get(l.contadorpregunta).id;
        String valor=et.getText().toString();
        Log.d("AppDolor", "texto escrito: "+valor);
        Utils.escribeRespuestaFichero("resultsinicial",preguntaid,valor);
        l.contadorpregunta=l.contadorpregunta+1;
        if (l.contadorpregunta<l.preguntas.size())
        {
            //siguiente pregunta
            Intent intent=new
            Intent(VisorPreguntaInicial.this,VisorPreguntaInicial.class);
            intent.putExtra("cont",l.contadorpregunta );
            VisorPreguntaInicial.this.startActivity(intent);
            VisorPreguntaInicial.this.finish();
        }
        else
        {
            //fin preguntas.
            l.contadorpregunta=0;
            new Wait2().execute("");
        }
    }
});

fondo.addView(rl);

}

if(p.tipo.equals("Slider"))
{

    final RelativeLayout rl= new RelativeLayout(this);
    rl.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
    LayoutParams.MATCH_PARENT));

```

```
rl.setPadding(5, 5, 5, 5);
```

```
TextView tv=new TextView(this);  
tv.setId(1);  
tv.setText(p.texto);  
tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,22);  
tv.setTextColor(Color.parseColor("#676767"));  
rl.addView(tv);
```

```
final TextView tv3=new TextView(getApplicationContext());  
tv3.setId(5);  
tv3.setText("");  
tv3.setTextSize(TypedValue.COMPLEX_UNIT_DIP,16);  
tv3.setTextColor(Color.parseColor("#2C88BF"));  
RelativeLayout.LayoutParams paramstv3= new  
RelativeLayout.LayoutParams(LayoutParams.WRAP_CONTENT ,  
LayoutParams.WRAP_CONTENT);  
paramstv3.setMargins(0,50,0,10);  
paramstv3.addRule(RelativeLayout.BELOW,tv.getId());  
paramstv3.addRule(RelativeLayout.CENTER_HORIZONTAL);  
tv3.setLayoutParams( paramstv3);  
rl.addView(tv3);
```

```
final SeekBar sb= new SeekBar(this);  
sb.setId(2);  
sb.setMax(p.numrespuestas-1);  
sb.setProgress(p.numrespuestas /2);  
sb.setProgressDrawable(getResources().getDrawable(R.drawable .progress_bar));
```

```
if (p.tiporespuesta ==1)  
{
```

```
    final TextView tv1=new TextView(this);  
    tv1.setId(3);  
    tv1.setText(p.respuestas .get(0).texto);  
    tv1.setTextSize(TypedValue.COMPLEX_UNIT_DIP,14);  
    tv1.setTextColor(Color.parseColor("#676767"));  
    RelativeLayout.LayoutParams paramstv1= new  
    RelativeLayout.LayoutParams(LayoutParams.WRAP_CONTENT ,  
    LayoutParams.WRAP_CONTENT);  
    paramstv1.setMargins(0,40,0,10);  
    paramstv1.addRule(RelativeLayout.BELOW,tv3.getId());  
    paramstv1.addRule(RelativeLayout.ALIGN_PARENT_LEFT);  
    tv1.setLayoutParams( paramstv1);  
    rl.addView(tv1);
```

```
    TextView tv2=new TextView(this);  
    tv2.setId(4);  
    tv2.setText(p.respuestas .get(p.numrespuestas-1).texto);  
    tv2.setTextSize(TypedValue.COMPLEX_UNIT_DIP,14);  
    tv2.setTextColor(Color.parseColor("#676767"));  
    RelativeLayout.LayoutParams paramstv2= new  
    RelativeLayout.LayoutParams(LayoutParams.WRAP_CONTENT ,  
    LayoutParams.WRAP_CONTENT);  
    paramstv2.setMargins(0,40,0,10);  
    paramstv2.addRule(RelativeLayout.BELOW,tv3.getId());  
    paramstv2.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);  
    tv2.setLayoutParams( paramstv2);  
    rl.addView(tv2);
```

```
    RelativeLayout.LayoutParams params= new  
    RelativeLayout.LayoutParams(LayoutParams.MATCH_PARENT ,  
    LayoutParams.WRAP_CONTENT);  
    params.setMargins(0,10,0,10);  
    params.addRule(RelativeLayout.BELOW,tv1.getId());
```

```
sb.setLayoutParams (params) ;

}

if (p.tiporespuesta ==2)
{
    DisplayMetrics metrics =
    getApplicationContext ().getResources ().getDisplayMetrics ();
    int width = metrics.widthPixels;
    int maximo= width/3;
    int tamaño=maximo - 150;

    if (tamaño>250)
    {
        tamaño=250;
    }

    if (tamaño<50)
    {
        tamaño=50;
    }

    final ImageView iv1=new ImageView(this) ;
    iv1.setId(6) ;
    iv1.setImageDrawable (Utils.LoadImageFromFile ( p.respuestas
    .get (0) .imagen )) ;
    RelativeLayout.LayoutParams paramsiv1= new
    RelativeLayout.LayoutParams (tamaño , tamaño) ;
    paramsiv1.setMargins (0,40,0,10) ;
    paramsiv1.addRule (RelativeLayout.BELOW,tv3.getId ()) ;
    paramsiv1.addRule (RelativeLayout.ALIGN_PARENT_LEFT) ;
    iv1.setLayoutParams ( paramsiv1) ;
    rl.addView (iv1) ;

    ImageView iv2=new ImageView(this) ;
    iv2.setId(7) ;
    iv2.setImageDrawable (Utils.LoadImageFromFile ( p.respuestas
    .get (p.numrespuestas /2) .imagen )) ;
    RelativeLayout.LayoutParams paramsiv2= new
    RelativeLayout.LayoutParams (tamaño , tamaño) ;
    paramsiv2.setMargins (0,40,0,10) ;
    paramsiv2.addRule (RelativeLayout.BELOW,tv3.getId ()) ;
    paramsiv2.addRule (RelativeLayout.CENTER_HORIZONTAL ) ;
    iv2.setLayoutParams ( paramsiv2) ;
    rl.addView (iv2) ;

    ImageView iv3=new ImageView(this) ;
    iv3.setId(8) ;
    iv3.setImageDrawable (Utils.LoadImageFromFile ( p.respuestas
    .get (p.numrespuestas-1) .imagen )) ;
    RelativeLayout.LayoutParams paramsiv3= new
    RelativeLayout.LayoutParams (tamaño , tamaño) ;
    paramsiv3.setMargins (0,40,0,10) ;
    paramsiv3.addRule (RelativeLayout.BELOW,tv3.getId ()) ;
    paramsiv3.addRule (RelativeLayout.ALIGN_PARENT_RIGHT) ;
    iv3.setLayoutParams ( paramsiv3) ;
    rl.addView (iv3) ;

    RelativeLayout.LayoutParams params= new
    RelativeLayout.LayoutParams (LayoutParams.MATCH_PARENT ,
    LayoutParams.WRAP_CONTENT) ;
    params.setMargins (0,10,0,10) ;
    params.addRule (RelativeLayout.BELOW,iv2.getId ()) ;
    sb.setLayoutParams (params) ;
}
rl.addView (sb) ;
```

```

sb.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onProgressChanged(SeekBar seekBar, int progress,boolean
fromUser) {
        // TODO Auto-generated method stub

        rl.removeView(tv3);
        int
valor=l.preguntas.get(l.contadorpregunta).respuestas.get(progress).val
or;
        tv3.setText("Valor seleccionado: "+ valor);
        rl.addView(tv3);

    }
});

Button bt = new Button(this);
bt.setText("OK");
bt.setTextSize(TypedValue.COMPLEX_UNIT_DIP,30);
bt.setBackgroundResource(R.drawable.botonazul);
bt.setTextColor(Color.WHITE);
RelativeLayout.LayoutParams params2= new
RelativeLayout.LayoutParams(MATCH_PARENT ,
LayoutParams.WRAP_CONTENT);
params2.setMargins(0,40,0,10);
params2.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM);
bt.setLayoutParams(params2);
rl.addView(bt);
bt.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        String preguntaid=l.preguntas.get(l.contadorpregunta).id;
        int valorsb=sb.getProgress();
        int
valor=l.preguntas.get(l.contadorpregunta).respuestas.get(valorsb).valo
r;
        Log.d("AppDolor", "slider seleccionado: "+valor);
        Utils.escribeRespuestaFichero("resultsinicial",preguntaid,valor);
        l.contadorpregunta=l.contadorpregunta+1;
        if (l.contadorpregunta<l.preguntas.size())
        {
            //siguiente pregunta
            Intent intent=new
Intent(VisorPreguntaInicial.this,VisorPreguntaInicial.class);
            intent.putExtra("cont",l.contadorpregunta );
            VisorPreguntaInicial.this.startActivity(intent);
            VisorPreguntaInicial.this.finish();
        }
        else
        {
            //fin preguntas.
            l.contadorpregunta=0;
            new Wait2().execute("");
        }

    }
});

```

```

        fondo.addView(r1);
    }

    if(p.tipo.equals("Check"))
    {

        TextView tv=new TextView(this);
        tv.setText(p.texto);
        tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP,22);
        tv.setTextColor(Color.parseColor("#676767"));
        fondo.addView(tv);

        TextView tv2=new TextView(this);
        tv2.setText("SELECCIONE LAS OPCIONES");
        tv2.setTextSize(TypedValue.COMPLEX_UNIT_DIP,14);
        tv2.setTextColor(Color.parseColor("#a1a9ab"));
        LinearLayout.LayoutParams params= new
        LinearLayout.LayoutParams(LayoutParams.WRAP_CONTENT ,
        LayoutParams.WRAP_CONTENT);
        params.setMargins(0,10,0,20);
        tv2.setLayoutParams(params);
        fondo.addView(tv2);

        ScrollView sv= new ScrollView(this);
        sv.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
        LayoutParams.WRAP_CONTENT));
        sv.setScrollbarFadingEnabled(false);

        LinearLayout ll= new LinearLayout(this);
        ll.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT ,
        LayoutParams.MATCH_PARENT));
        ll.setPadding(5, 5, 5, 5);
        ll.setOrientation(LinearLayout.VERTICAL );

        final ToggleButton[] botones= new ToggleButton[p.numrespuestas];

        DisplayMetrics metrics =
        getApplicationContext().getResources().getDisplayMetrics();
        int width=metrics.widthPixels;
        float aspectRatio=800/(float)118;
        int altoboton=Math.round(width/aspectRatio);

        for (int i=0; i<p.respuestas.size(); i++)
        {

            botones[i] = new ToggleButton(this);

            if (p.tiporespuesta==1)
            {
                botones[i].setText(p.respuestas .get(i).texto);
                botones[i].setTextOff(p.respuestas .get(i).texto);
                botones[i].setTextOn(p.respuestas .get(i).texto);
            }

            if (p.tiporespuesta ==2)
            {
                ImageSpan imageSpan = new
                ImageSpan(this,Utills.LoadBitmapFromFileWithDimensions(p.respuestas
                .get(i).imagen ,altoboton));
                SpannableString content = new SpannableString("X");
                content.setSpan(imageSpan, 0,1,Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
                botones[i].setText(content);
                botones[i].setTextOff(content);
                botones[i].setTextOn(content);
            }
        }
    }

```

```

if (p.tiporespuesta ==3)
{
    ImageSpan imageSpan = new
    ImageSpan(this,Utils.LoadBitmapFromFileWithDimensions (p.respuestas
    .get(i).imagen ,altoboton));
    String texto=p.respuestas .get(i).texto;
    SpannableString content = new SpannableString("X "+texto);
    content.setSpan (imageSpan, 0,1,Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
    botones[i].setText (content);
    botones[i].setTextOff (content);
    botones[i].setTextOn (content);
}

botones[i].setBackgroundResource (R.drawable.botones_style );
botones[i].setTextSize (TypedValue.COMPLEX_UNIT_DIP,18);
botones[i].setTextColor (Color.parseColor ("#676767"));
botones[i].setId (i);
botones[i].setGravity (Gravity.LEFT|Gravity.CENTER);
botones[i].setPadding (20, 0, 75,0);
ll.addView (botones [i]);
botones [i].setOnCheckedChangeListener (new
CompoundButton.OnCheckedChangeListener () {

    @Override
    public void onCheckedChanged (CompoundButton buttonView, boolean
    isChecked) {
        if (isChecked)
        {
            int indice=buttonView.getId ();
            botones [indice].setTextColor (Color.parseColor ("#2C88BF"));
        }
        else
        {
            int indice=buttonView.getId ();
            botones [indice].setTextColor (Color.parseColor ("#676767"));
        }
    }
});
}

Button salir= new Button (this);
salir.setText ("OK");
salir.setTextSize (TypedValue.COMPLEX_UNIT_DIP,30);
salir.setBackgroundResource (R.drawable.botonazul);
salir.setTextColor (Color.WHITE);
LinearLayout.LayoutParams params2= new
LinearLayout.LayoutParams (LinearLayout.LayoutParams.MATCH_PARENT ,
LinearLayout.LayoutParams.WRAP_CONTENT);
params2.setMargins (0,50,0,0);
salir.setLayoutParams (params2);
ll.addView (salir);

salir.setOnClickListener (new View.OnClickListener () {

    @Override
    public void onClick (View v) {

        String preguntaid=l.preguntas.get (l.contadorpregunta).id;
        List<Integer> marcados=new ArrayList<Integer> ();
        for (int i=0; i<botones.length; i++)
        {
            if (botones [i].isChecked ())
            {
                marcados.add (i);
            }
        }

        if (marcados.size ()>0)
        {

```

```

String valor="";
for (int marcado:marcados)
{
    if (valor.isEmpty())
    {

        valor=Integer.toString(l.preguntas.get(l.contadorpr
egunta).respuestas .get(marcado).valor);
    }
    else
    {
        valor=valor+"*"+Integer.toString(
l.preguntas.get(l.contadorpregunta).respuestas
.get(marcado).valor);
    }
}
Log.d("AppPruebaUsabilidad", "valor seleccionado:
"+valor);

Utils.escribeRespuestaFichero("resultsinicial",preguntaid,
valor);
l.contadorpregunta=l.contadorpregunta+1;
if (l.contadorpregunta<l.preguntas.size())
{
    //siguiente pregunta
    Intent intent=new
Intent(VisorPreguntaInicial.this,VisorPreguntaInicial.
class);
    intent.putExtra("cont",l.contadorpregunta );
    VisorPreguntaInicial.this.startActivity(intent);
    VisorPreguntaInicial.this.finish();
}
else
{
    //fin preguntas.
    l.contadorpregunta=0;
    new Wait2().execute("");
}
}

});

sv.addView(ll);

fondo.addView(sv);
}

}
}

private class Wait2 extends AsyncTask<String, Void, Integer> {

protected void onPreExecute() {
// pb2.setVisibility(View.VISIBLE);
Utils.cerrarFicheroResultados("resultsinicial");
SharedPreferences settings =
getApplicationContext().getSharedPreferences(Utils.appname, 0);
SharedPreferences.Editor editor = settings.edit();
editor.putBoolean("inicial", true);
editor.commit();
}

protected Integer doInBackground(String... urls) {

```

```
if (Utils.isOnline(getApplicationContext()))
{
    //politica de seguridad
    StrictMode.ThreadPolicy policy = new
    StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);

    Utils.almacenarResultados("resultsinicial", "2");
}

return null;
}

protected void onPostExecute(Integer valor) {
// pb2.setVisibility(View.GONE);
Intent intent=new Intent(VisorPreguntaInicial.this,CambioHora.class);
intent.putExtra("origen",1 );
VisorPreguntaInicial .this.startActivity(intent);
VisorPreguntaInicial .this.finish();
}
}
}
```

```
<LinearLayout 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:orientation="vertical"
    android:background="#FFFFFF"
    tools:context=".CambioHora" >

    <!--
        The primary full-screen view. This can be replaced with whatever view
        is needed to present your content, e.g. VideoView, SurfaceView,
        TextureView, etc.
    -->

    <TextView
        android:id="@+id/titulo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="30dp"
        android:layout_alignParentTop="true"
        android:textSize="24dp"
        android:text="Configuración del horario"
        android:textColor="#676767" />

    <TextView
        android:id="@+id/titulo2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="10dp"
        android:paddingBottom="20dp"
        android:layout_alignParentTop="true"
        android:textSize="14dp"
        android:text="ESTA APLICACIÓN LE REALIZARÁ PREGUNTAS EN UN HORARIO
ESTABLECIDO DURANTE UN TIEMPO. USTED PUEDE VARIAR DICHO HORARIO UN MÁXIMO DE
2 HORAS"
        android:textColor="#a1a9ab" />

    <ProgressBar style="?android:attr/progressBarStyleLarge"
        android:id="@+id/progressBar1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal">

        </ProgressBar>
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <LinearLayout
            android:id="@+id/layouthora"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="fill_horizontal"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:orientation="vertical" >

            </LinearLayout>

    </ScrollView>
</LinearLayout>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/layoutfondo"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:orientation="vertical" >

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="fitXY"
        android:src="@drawable/cabe2" >
    </ImageView>

    <RelativeLayout
        android:id="@+id/layoutinstalacion1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="5dp"
        android:paddingRight="5dp" >

        <TextView
            android:id="@+id/menasjeinstalacion"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:gravity="center"
            android:text="Muchas gracias por instalar la aplicación, por favor espere
            mientras se configura"
            android:textColor="#2C88BF"
            android:textSize="24dp" />

        <ProgressBar
            android:id="@+id/pbinstalacion"
            style="?android:attr/progressBarStyleLarge"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true" >
        </ProgressBar>
    </RelativeLayout>

    <RelativeLayout
        android:id="@+id/layoutinstalacion2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="5dp"
        android:paddingRight="5dp" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:paddingLeft="5dp"
            android:paddingRight="5dp" >

            <TextView
                android:id="@+id/menasjeinstalacion2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Por favor responde a las siguiente preguntas"
                android:textColor="#2C88BF"
                android:textSize="20dp" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:paddingLeft="5dp"
```

```
android:paddingRight="5dp"
android:paddingTop="10dp" >
```

```
<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="3"
    android:layout_gravity="center_vertical"
    android:text="Fecha de Nacimiento"
    android:textSize="16dp" />
```

```
<EditText
    android:id="@+id/fecha"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:editable="false"
    android:textSize="16dp" />
```

```
<ImageButton
    android:id="@+id/calendar"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:layout_weight="1"
    android:src="@drawable/calendario"
    android:scaleType="fitCenter"
    android:adjustViewBounds="true">
```

```
</ImageButton>
</LinearLayout>
```

```
</LinearLayout>
```

```
<Button
    android:id="@+id/instalacionaceptar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:background="@drawable/botonazul"
    android:text="Aceptar"
    android:textColor="#FFFFFF"
    android:textSize="30dp" />
```

```
<ProgressBar
    android:id="@+id/pbinstalacion2"
    style="?android:attr/progressBarStyleLarge"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true">
```

```
</ProgressBar>
</RelativeLayout>
```

```
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layoutfondo"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#FFFFFF" >

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="fitXY"
        android:src="@drawable/cabe2" >
    </ImageView>

</LinearLayout>
```

```
<LinearLayout 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:orientation="vertical"
    android:background="#FFFFFF">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="fitXY"
        android:src="@drawable/cabe2" >
    </ImageView>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="5dp"
        android:paddingRight="5dp" >

        <LinearLayout
            android:id="@+id/layoutinstrucciones"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/menasjeinstalacion2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerHorizontal="true"
                android:gravity="center"
                android:textSize="20dp"
                android:text="INSTRUCCIONES"
                android:textColor="#2C88BF" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="16dp"
                android:paddingTop="10dp"
                android:text="1. Rellenar la encuesta acerca de las características de tu
                problema" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="16dp"
                android:paddingTop="10dp"
                android:text="2. Durante 30 días responder 2 veces al día (mañana y tarde) a
                unas breves preguntas (dolor, cansancio, estado de ánimo, etc)" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="16dp"
                android:paddingTop="10dp"
                android:text="3. MUY IMPORTANTE. Cuando tengas un episodio de dolor, entra
                en la aplicación y responde a las preguntas." />
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:textSize="16dp"
                    android:paddingTop="10dp"
                    android:text="4. Si tienes cualquier duda puedes escribirnos a
                    dolorcronico@uji.es" />

        </LinearLayout>

    </RelativeLayout>

</LinearLayout>
```

<Button

```
    android:id="@+id/instruccionesok"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_marginBottom="10dp"  
    android:text="OK"  
    android:textSize="30dp"  
    android:textColor="#FFFFFF"  
    android:background="@drawable/botonazul" />
```

</RelativeLayout>

</LinearLayout>

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

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/layoutfondo"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:background="#FFFFFF" >
```

```
    <ImageView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:scaleType="fitXY"  
        android:src="@drawable/cabecera2" >  
</ImageView>
```

```
<RelativeLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:paddingLeft="5dp"  
    android:paddingRight="5dp" >
```

```
<TextView  
    android:id="@+id/instrucciones"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingTop="30dp"  
    android:layout_alignParentTop="true"  
    android:textSize="24dp"  
    android:text="Muchas gracias por instalar la aplicación. Por favor introduce  
    tu usuario"  
    android:textColor="#676767" />
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:layout_below="@+id/instrucciones"  
    android:paddingTop="30dp" >
```

```
    <EditText  
        android:id="@+id/login"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:background="@drawable/cajaidpaciente"  
        android:textColor="#b6b6b6"  
        android:paddingLeft="10dp" />
```

```
</LinearLayout>
```

```
<Button  
    android:id="@+id/continuar"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_marginBottom="10dp"  
    android:text="Aceptar"  
    android:textSize="30dp"  
    android:textColor="#FFFFFF"  
    android:background="@drawable/botonazul" />
```

```
</RelativeLayout>
```

```
</LinearLayout>
```

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

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:background="@drawable/fondo">
```

```
    <ProgressBar style="?android:attr/progressBarStyleLarge"  
        android:id="@+id/progressBar1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_centerInParent="true">
```

```
</ProgressBar>
```

```
</RelativeLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
  <item
    android:id="@+id/action_exit"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/action_exit"/>
</menu>
```

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
  <item
    android:id="@+id/action_back"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/action_back"/>
</menu>
```

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

  <!--      <item
    android:id="@+id/action_settings"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/action_settings"/> -->

  <item
    android:id="@+id/action_exit"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/action_exit"/>

</menu>
```

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

  <item
    android:id="@+id/action_back"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/action_back"/>

</menu>
```

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

  <item
    android:id="@+id/action_settings"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/action_settings"/>

  <item
    android:id="@+id/action_exit"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/action_exit"/>

</menu>
```

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
  <item
    android:id="@+id/action_instructions"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/action_instructions"/>

  <item
    android:id="@+id/action_settings"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/action_settings"/>

</menu>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <inicial>
    <pregunta>
      <id>D02_Usuario</id>
      <texto>¿Qué tipo de usuario eres? </texto>
      <numeroRespuestas>2</numeroRespuestas>
      <tipo>Radio</tipo>
      <tiporespuesta>1</tiporespuesta>
      <respuesta>
        <textorespuesta>Soy una persona con dolor crónico</textorespuesta>
        <valor>1</valor>
      </respuesta>
      <respuesta>
        <textorespuesta>NO tengo dolor crónico, pero quiero ver la
          aplicación.</textorespuesta>
        <valor>2</valor>
      </respuesta>
    </pregunta>
  </pregunta>
  <pregunta>
    <id>D03_Genero</id>
    <texto>Indica tu género: </texto>
    <numeroRespuestas>2</numeroRespuestas>
    <tipo>Radio</tipo>
    <tiporespuesta>1</tiporespuesta>
    <respuesta>
      <textorespuesta>Hombre</textorespuesta>
      <valor>0</valor>
    </respuesta>
    <respuesta>
      <textorespuesta>Mujer</textorespuesta>
      <valor>1</valor>
    </respuesta>
  </pregunta>
  <pregunta>
    <id>D04_TipoDolor</id>
    <texto>Indica tu tipo de dolor. Puedes marcar mas de una respuesta:</texto>
    <numeroRespuestas>8</numeroRespuestas>
    <tipo>Check</tipo>
    <tiporespuesta>1</tiporespuesta>
    <respuesta>
      <textorespuesta>Fibromialgia</textorespuesta>
      <valor>1</valor>
    </respuesta>
    <respuesta>
      <textorespuesta>Lumbalgia</textorespuesta>
      <valor>2</valor>
    </respuesta>
    <respuesta>
      <textorespuesta>Cervicalgia</textorespuesta>
      <valor>3</valor>
    </respuesta>
    <respuesta>
      <textorespuesta>Artritis</textorespuesta>
      <valor>4</valor>
    </respuesta>
    <respuesta>
      <textorespuesta>Artrosis</textorespuesta>
      <valor>5</valor>
    </respuesta>
    <respuesta>
      <textorespuesta>Cefaleas</textorespuesta>
      <valor>6</valor>
    </respuesta>
    <respuesta>
      <textorespuesta>Dolor neuropatico</textorespuesta>
      <valor>7</valor>
    </respuesta>
    <respuesta>
      <textorespuesta>Otro tipo de dolor</textorespuesta>
      <valor>8</valor>
    </respuesta>
  </pregunta>
</config>
```

```

    </respuesta>
</pregunta>
<pregunta>
  <id>D05_TipoDolorOtro</id>
  <texto>Si has seleccionado "Otro tipo de dolor", escribe cuál. Si NO, deja en
  blanco esta pregunta. Cuando termines pulsa OK.</texto>
  <numeroRespuestas>1</numeroRespuestas>
  <tipo>Texto</tipo>
</pregunta>
<pregunta>
  <id>D06_LocalizacionDolor</id>
  <texto>Indica en qué parte del cuerpo es más intenso tu dolor. Puedes marcar mas
  de una respuesta:</texto>
  <numeroRespuestas>16</numeroRespuestas>
  <tipo>Check</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Cabeza</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Hombros</textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Cervicales</textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Espalda alta (dorsales)</textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Espalda baja (lumbares)</textorespuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Brazos</textorespuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Muñecas</textorespuesta>
    <valor>7</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Manos</textorespuesta>
    <valor>8</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Abdomen</textorespuesta>
    <valor>9</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Torax</textorespuesta>
    <valor>10</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Caderas</textorespuesta>
    <valor>11</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Piernas</textorespuesta>
    <valor>12</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Rodillas</textorespuesta>
    <valor>13</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Pies</textorespuesta>
    <valor>14</valor>

```

```

</respuesta>
<respuesta>
  <textorespuesta>Todo el cuerpo</textorespuesta>
  <valor>15</valor>
</respuesta>
<respuesta>
  <textorespuesta>Otra parte distinta</textorespuesta>
  <valor>16</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D07_DiagnosticoDolor</id>
  <texto>¿Quién te diagnosticó tu problema de dolor?</texto>
  <numeroRespuestas>10</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Médico de familia</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Reumatólogo</textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Traumatólogo</textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Rehabilitador</textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Psiquiatra</textorespuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Unidad del Dolor</textorespuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Neurocirujano</textorespuesta>
    <valor>7</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Neurólogo</textorespuesta>
    <valor>8</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Oncólogo</textorespuesta>
    <valor>9</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Otro profesional</textorespuesta>
    <valor>10</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D08_DuracionDolor</id>
  <texto>Indica cuánto hace que dura tu dolor:</texto>
  <numeroRespuestas>4</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Menos de 1 año</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Entre 1 y 5 años</textorespuesta>
    <valor>2</valor>
  </respuesta>

```

```

<respuesta>
  <textorespuesta>Más de 5 años y menos de 10 años</textorespuesta>
  <valor>3</valor>
</respuesta>
<respuesta>
  <textorespuesta>Más de 10 años</textorespuesta>
  <valor>4</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D09_TratamientoActual</id>
  <texto>¿Sigues actualmente alguno de estos tratamientos para el dolor? Puedes
  marcar mas de uno</texto>
  <numeroRespuestas>6</numeroRespuestas>
  <tipo>Check</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Fisioterapia</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Tratamiento farmacológico</textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Infiltraciones</textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Tratamiento psicológico</textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Tratamientos naturales/alternativos</textorespuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>No sigo ningún tratamiento para el dolor</textorespuesta>
    <valor>6</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D10_TratamientoNuevo</id>
  <texto>En el último mes, ¿has empezado un nuevo tratamiento para el dolor
  crónico?</texto>
  <numeroRespuestas>2</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Sí</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>No</textorespuesta>
    <valor>0</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D11_TratamientoNuevoCual</id>
  <texto>Selecciona cuál de estos tratamientos has empezado en el último mes.
  Puedes marcar mas de uno</texto>
  <numeroRespuestas>6</numeroRespuestas>
  <tipo>Check</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Fisioterapia</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Tratamiento farmacológico</textorespuesta>
    <valor>2</valor>

```

```

</respuesta>
<respuesta>
  <textorespuesta>Infiltraciones</textorespuesta>
  <valor>3</valor>
</respuesta>
<respuesta>
  <textorespuesta>Tratamiento psicológico</textorespuesta>
  <valor>4</valor>
</respuesta>
<respuesta>
  <textorespuesta>Tratamientos naturales/alternativos</textorespuesta>
  <valor>5</valor>
</respuesta>
<respuesta>
  <textorespuesta>No he empezado ninguno</textorespuesta>
  <valor>6</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D12_EstadoCivil</id>
  <texto>Indica tu estado civil:</texto>
  <numeroRespuestas>6</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Soltero/a</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Casado/a</textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>En pareja</textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Divorciado/a</textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Separado/a</textorespuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Viudo/a</textorespuesta>
    <valor>6</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D13_EstadoLaboral</id>
  <texto>Indica tu situación laboral:</texto>
  <numeroRespuestas>7</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Trabajador/a activo/a</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Baja temporal</textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Incapacidad permanente</textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>En paro</textorespuesta>
    <valor>4</valor>
  </respuesta>

```

```

<respuesta>
  <textorespuesta>Ama/o de casa</textorespuesta>
  <valor>5</valor>
</respuesta>
<respuesta>
  <textorespuesta>Jubilado/a</textorespuesta>
  <valor>6</valor>
</respuesta>
<respuesta>
  <textorespuesta>Estudiante</textorespuesta>
  <valor>7</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D14_Estudios</id>
  <texto>Indica tu nivel de estudios superados:</texto>
  <numeroRespuestas>5</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Sin estudios</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Estudios Primarios/EGB</textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Estudios Secundarios/Bachiller</textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Ciclo Superior/Formación Profesional</textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Estudios Universitarios</textorespuesta>
    <valor>5</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D15_DiagnosticoDepresion</id>
  <texto>Actualmente, ¿sufres de depresión diagnosticada por un médico o psicólogo?</texto>
  <numeroRespuestas>2</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Sí</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>No</textorespuesta>
    <valor>0</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D16_DiagnosticoAnsiedad</id>
  <texto>Actualmente, ¿sufres de ansiedad diagnosticada por un médico o psicólogo?</texto>
  <numeroRespuestas>2</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Sí</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>No</textorespuesta>
    <valor>0</valor>
  </respuesta>

```

```
</pregunta>  
</inicial>  
</config>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <maxduracion>30</maxduracion>
  <demanda>
    <duracion>30</duracion>
    <pregunta>
      <id>D17_Dolor</id>
      <texto>¿Qué intensidad de DOLOR sientes AHORA? </texto>
      <numeroRespuestas>11</numeroRespuestas>
      <tipo>Slider</tipo>
      <tiporespuesta>1</tiporespuesta>
      <respuesta>
        <textorespuesta>0=Ningún dolor</textorespuesta>
        <valor>0</valor>
      </respuesta>
      <respuesta>
        <valor>1</valor>
      </respuesta>
      <respuesta>
        <valor>2</valor>
      </respuesta>
      <respuesta>
        <valor>3</valor>
      </respuesta>
      <respuesta>
        <valor>4</valor>
      </respuesta>
      <respuesta>
        <valor>5</valor>
      </respuesta>
      <respuesta>
        <valor>6</valor>
      </respuesta>
      <respuesta>
        <valor>7</valor>
      </respuesta>
      <respuesta>
        <valor>8</valor>
      </respuesta>
      <respuesta>
        <valor>9</valor>
      </respuesta>
      <respuesta>
        <textorespuesta>10=Dolor extremo</textorespuesta>
        <valor>10</valor>
      </respuesta>
    </pregunta>
    <pregunta>
      <id>D18_Cansancio</id>
      <texto>¿Qué intensidad de CANSANCIO sientes AHORA? </texto>
      <numeroRespuestas>11</numeroRespuestas>
      <tipo>Slider</tipo>
      <tiporespuesta>1</tiporespuesta>
      <respuesta>
        <textorespuesta>0=Ningún cansancio</textorespuesta>
        <valor>0</valor>
      </respuesta>
      <respuesta>
        <valor>1</valor>
      </respuesta>
      <respuesta>
        <valor>2</valor>
      </respuesta>
      <respuesta>
        <valor>3</valor>
      </respuesta>
      <respuesta>
        <valor>4</valor>
      </respuesta>
      <respuesta>
        <valor>5</valor>
      </respuesta>
    </pregunta>
  </demanda>
</config>
```



```
</respuesta>
<respuesta>
  <valor>6</valor>
</respuesta>
<respuesta>
  <valor>7</valor>
</respuesta>
<respuesta>
  <valor>8</valor>
</respuesta>
<respuesta>
  <valor>9</valor>
</respuesta>
<respuesta>
  <textorespuesta>10=Cansancio extremo</textorespuesta>
  <valor>10</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D19_Alegría</id>
  <texto>¿Qué intensidad de ALEGRÍA O BIENESTAR sientes AHORA?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Nada alegre</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <valor>7</valor>
  </respuesta>
  <respuesta>
    <valor>8</valor>
  </respuesta>
  <respuesta>
    <valor>9</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>10=Totalmente alegre</textorespuesta>
    <valor>10</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D20 Tristeza</id>
  <texto>¿Qué intensidad de TRISTEZA O ESTADO DE ÁNIMO BAJO sientes AHORA?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Nada triste</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
```

```
<valor>1</valor>
</respuesta>
<respuesta>
  <valor>2</valor>
</respuesta>
<respuesta>
  <valor>3</valor>
</respuesta>
<respuesta>
  <valor>4</valor>
</respuesta>
<respuesta>
  <valor>5</valor>
</respuesta>
<respuesta>
  <valor>6</valor>
</respuesta>
<respuesta>
  <valor>7</valor>
</respuesta>
<respuesta>
  <valor>8</valor>
</respuesta>
<respuesta>
  <valor>9</valor>
</respuesta>
<respuesta>
  <textorespuesta>10=Totalmente triste</textorespuesta>
  <valor>10</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D21_Ansiedad</id>
  <texto>¿Qué intensidad de ANSIEDAD O NERVIOSISMO sientes AHORA?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Nada
Ansioso/a</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <valor>7</valor>
  </respuesta>
  <respuesta>
    <valor>8</valor>
  </respuesta>
  <respuesta>
    <valor>9</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>10=Totalmente
```

```
Ansioso/a</textorespuesta>
  <valor>10</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D41_Enfado</id>
  <texto>¿Qué intensidad de ENFADO sientes AHORA?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Nada enfadado/a</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <valor>7</valor>
  </respuesta>
  <respuesta>
    <valor>8</valor>
  </respuesta>
  <respuesta>
    <valor>9</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>10=Totalmente enfadado/a</textorespuesta>
    <valor>10</valor>
  </respuesta>
</pregunta>
</demanda>
<lanzar>
  <hora>11:00:00</hora>
  <duracion>30</duracion>
  <pregunta>
    <id>D17_Dolor</id>
    <texto>¿Qué intensidad de DOLOR sientes AHORA? </texto>
    <numeroRespuestas>11</numeroRespuestas>
    <tipo>Slider</tipo>
    <tiporespuesta>1</tiporespuesta>
    <respuesta>
      <textorespuesta>0=Ningún dolor</textorespuesta>
      <valor>0</valor>
    </respuesta>
    <respuesta>
      <valor>1</valor>
    </respuesta>
    <respuesta>
      <valor>2</valor>
    </respuesta>
    <respuesta>
      <valor>3</valor>
    </respuesta>
    <respuesta>
      <valor>4</valor>
```

```
</respuesta>
<respuesta>
  <valor>5</valor>
</respuesta>
<respuesta>
  <valor>6</valor>
</respuesta>
<respuesta>
  <valor>7</valor>
</respuesta>
<respuesta>
  <valor>8</valor>
</respuesta>
<respuesta>
  <valor>9</valor>
</respuesta>
<respuesta>
  <textorespuesta>10=Dolor extremo</textorespuesta>
  <valor>10</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D18_Cansancio</id>
  <texto>¿Qué intensidad de CANSANCIO sientes AHORA? </texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Ningún cansancio</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <valor>7</valor>
  </respuesta>
  <respuesta>
    <valor>8</valor>
  </respuesta>
  <respuesta>
    <valor>9</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>10=Cansancio extremo</textorespuesta>
    <valor>10</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D19_Alegría</id>
  <texto>¿Qué intensidad de ALEGRÍA O BIENESTAR sientes AHORA?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Nada alegre</textorespuesta>
```

```
<valor>0</valor>
</respuesta>
<respuesta>
  <valor>1</valor>
</respuesta>
<respuesta>
  <valor>2</valor>
</respuesta>
<respuesta>
  <valor>3</valor>
</respuesta>
<respuesta>
  <valor>4</valor>
</respuesta>
<respuesta>
  <valor>5</valor>
</respuesta>
<respuesta>
  <valor>6</valor>
</respuesta>
<respuesta>
  <valor>7</valor>
</respuesta>
<respuesta>
  <valor>8</valor>
</respuesta>
<respuesta>
  <valor>9</valor>
</respuesta>
<respuesta>
  <textorespuesta>10=Totalmente alegre</textorespuesta>
  <valor>10</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D20 Tristeza</id>
  <texto>¿Qué intensidad de TRISTEZA O ESTADO DE ÁNIMO BAJO sientes AHORA?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Nada triste</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <valor>7</valor>
  </respuesta>
  <respuesta>
    <valor>8</valor>
  </respuesta>
  <respuesta>
    <valor>9</valor>
  </respuesta>
</pregunta>
```

```
<respuesta>
  <textorespuesta>10=Totalmente triste</textorespuesta>
  <valor>10</valor>
</respuesta>
```

```
</pregunta>
```

```
<pregunta>
```

```
<id>D21_Ansiedad</id>
```

```
<texto>¿Qué intensidad de ANSIEDAD O NERVIOSISMO sientes AHORA?</texto>
```

```
<numeroRespuestas>11</numeroRespuestas>
```

```
<tipo>Slider</tipo>
```

```
<tiporespuesta>1</tiporespuesta>
```

```
<respuesta>
```

```
  <textorespuesta>0=Nada
```

```
Ansioso/a</textorespuesta>
```

```
  <valor>0</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>1</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>2</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>3</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>4</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>5</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>6</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>7</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>8</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>9</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <textorespuesta>10=Totalmente
```

```
Ansioso/a</textorespuesta>
```

```
  <valor>10</valor>
```

```
</respuesta>
```

```
</pregunta>
```

```
<pregunta>
```

```
<id>D41_Enfado</id>
```

```
<texto>¿Qué intensidad de ENFADO sientes AHORA?</texto>
```

```
<numeroRespuestas>11</numeroRespuestas>
```

```
<tipo>Slider</tipo>
```

```
<tiporespuesta>1</tiporespuesta>
```

```
<respuesta>
```

```
  <textorespuesta>0=Nada enfadado/a</textorespuesta>
```

```
  <valor>0</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>1</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>2</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>3</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
  <valor>4</valor>
```

```
</respuesta>
```

```
<respuesta>
  <valor>5</valor>
</respuesta>
<respuesta>
  <valor>6</valor>
</respuesta>
<respuesta>
  <valor>7</valor>
</respuesta>
<respuesta>
  <valor>8</valor>
</respuesta>
<respuesta>
  <valor>9</valor>
</respuesta>
<respuesta>
  <textorespuesta>10=Totalmente enfadado/a</textorespuesta>
  <valor>10</valor>
</respuesta>
```

```
</pregunta>
```

```
<pregunta>
```

```
<id>D22_SaludGeneral</id>
```

```
<texto>En general, dirías que tu SALUD es:</texto>
```

```
<numeroRespuestas>5</numeroRespuestas>
```

```
<tipo>Radio</tipo>
```

```
<tiporespuesta>1</tiporespuesta>
```

```
<respuesta>
```

```
<textorespuesta>Muy mala </textorespuesta>
```

```
<valor>1</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<textorespuesta>Mala </textorespuesta>
```

```
<valor>2</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<textorespuesta>Ni buena, ni mala </textorespuesta>
```

```
<valor>3</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<textorespuesta>Buena</textorespuesta>
```

```
<valor>4</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<textorespuesta>Muy buena</textorespuesta>
```

```
<valor>5</valor>
```

```
</respuesta>
```

```
</pregunta>
```

```
<pregunta>
```

```
<id>D23_InterferenciaDormir</id>
```

```
<texto>¿Ha interferido el DOLOR en la calidad de tu SUEÑO esta noche?</texto>
```

```
<numeroRespuestas>11</numeroRespuestas>
```

```
<tipo>Slider</tipo>
```

```
<tiporespuesta>1</tiporespuesta>
```

```
<respuesta>
```

```
<textorespuesta>0=Ninguna
```

```
Interferencia</textorespuesta>
```

```
<valor>0</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>1</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>2</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>3</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>4</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
    <valor>5</valor>
  </respuesta>
</respuesta>
    <valor>6</valor>
  </respuesta>
</respuesta>
    <valor>7</valor>
  </respuesta>
</respuesta>
    <valor>8</valor>
  </respuesta>
</respuesta>
    <valor>9</valor>
  </respuesta>
</respuesta>
    <textorespuesta>10=Máxima
```

Interferencia</textorespuesta>

```
    <valor>10</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D24_EvitacionTrabajo</id>
  <texto>Indica tu grado de acuerdo con esta afirmación: Con mi dolor actual, no debería hacer mi trabajo habitual (incluye trabajo en casa o fuera).</texto>
  <numeroRespuestas>5</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Nada de acuerdo </textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Poco de acuerdo </textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Ni de acuerdo, ni en desacuerdo </textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Bastante de acuerdo </textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Totalmente de acuerdo</textorespuesta>
    <valor>5</valor>
  </respuesta>
```

```
</pregunta>
<pregunta>
  <id>D25_Desesperanza</id>
  <texto>Indica tu grado de acuerdo con esta afirmación: Tener dolor es horrible y siento que el dolor es más fuerte que yo</texto>
  <numeroRespuestas>5</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Nada de acuerdo </textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Poco de acuerdo </textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Ni de acuerdo, ni en desacuerdo </textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Bastante de acuerdo </textorespuesta>
    <valor>4</valor>
  </respuesta>
```

```
<respuesta>
  <textorespuesta>Totalmente de acuerdo</textorespuesta>
  <valor>5</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D26_Willingness</id>
  <texto>Indica tu grado de acuerdo con esta afirmación: Antes de poder hacer planes en serio, tengo que tener algo de control sobre el dolor</texto>
  <numeroRespuestas>5</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Nada de acuerdo </textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Poco de acuerdo </textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Ni de acuerdo, ni en desacuerdo </textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Bastante de acuerdo </textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Totalmente de acuerdo</textorespuesta>
    <valor>5</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D27_EvitacionEjercicio</id>
  <texto>Indica tu grado de acuerdo con esta afirmación: La actividad física hace que mi dolor empeore. </texto>
  <numeroRespuestas>5</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Nada de acuerdo </textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Poco de acuerdo </textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Ni de acuerdo, ni en desacuerdo </textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Bastante de acuerdo </textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Totalmente de acuerdo </textorespuesta>
    <valor>5</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D28_Aceptación</id>
  <texto>Para finalizar, indica tu grado de acuerdo con esta afirmación: Estoy viviendo una vida plena aunque tenga dolor crónico</texto>
  <numeroRespuestas>5</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Nada de acuerdo </textorespuesta>
    <valor>1</valor>
```

```
</respuesta>
<respuesta>
  <textorespuesta>Poco de acuerdo </textorespuesta>
  <valor>2</valor>
</respuesta>
<respuesta>
  <textorespuesta>Ni de acuerdo, ni en desacuerdo </textorespuesta>
  <valor>3</valor>
</respuesta>
<respuesta>
  <textorespuesta>Bastante de acuerdo </textorespuesta>
  <valor>4</valor>
</respuesta>
<respuesta>
  <textorespuesta>Totalmente de acuerdo</textorespuesta>
  <valor>5</valor>
</respuesta>
</pregunta>
</lanzar>
<lanzar>
  <hora>19:00:00</hora>
  <duracion>30</duracion>
  <pregunta>
    <id>D17_Dolor</id>
    <texto>¿Qué intensidad de DOLOR sientes AHORA? </texto>
    <numeroRespuestas>11</numeroRespuestas>
    <tipo>Slider</tipo>
    <tiporespuesta>1</tiporespuesta>
    <respuesta>
      <textorespuesta>0=Ningún dolor</textorespuesta>
      <valor>0</valor>
    </respuesta>
    <respuesta>
      <valor>1</valor>
    </respuesta>
    <respuesta>
      <valor>2</valor>
    </respuesta>
    <respuesta>
      <valor>3</valor>
    </respuesta>
    <respuesta>
      <valor>4</valor>
    </respuesta>
    <respuesta>
      <valor>5</valor>
    </respuesta>
    <respuesta>
      <valor>6</valor>
    </respuesta>
    <respuesta>
      <valor>7</valor>
    </respuesta>
    <respuesta>
      <valor>8</valor>
    </respuesta>
    <respuesta>
      <valor>9</valor>
    </respuesta>
    <respuesta>
      <textorespuesta>10=Dolor extremo</textorespuesta>
      <valor>10</valor>
    </respuesta>
  </pregunta>
  <pregunta>
    <id>D18_Cansancio</id>
    <texto>¿Qué intensidad de CANSANCIO sientes AHORA? </texto>
    <numeroRespuestas>11</numeroRespuestas>
    <tipo>Slider</tipo>
    <tiporespuesta>1</tiporespuesta>
    <respuesta>
```

```
<textorespuesta>0=Ningún cansancio</textorespuesta>
<valor>0</valor>
</respuesta>
<respuesta>
  <valor>1</valor>
</respuesta>
<respuesta>
  <valor>2</valor>
</respuesta>
<respuesta>
  <valor>3</valor>
</respuesta>
<respuesta>
  <valor>4</valor>
</respuesta>
<respuesta>
  <valor>5</valor>
</respuesta>
<respuesta>
  <valor>6</valor>
</respuesta>
<respuesta>
  <valor>7</valor>
</respuesta>
<respuesta>
  <valor>8</valor>
</respuesta>
<respuesta>
  <valor>9</valor>
</respuesta>
<textorespuesta>10=Cansancio extremo</textorespuesta>
<valor>10</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D19_Alegría</id>
  <texto>¿Qué intensidad de ALEGRÍA O BIENESTAR sientes AHORA?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Nada alegre</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <valor>7</valor>
  </respuesta>
  <respuesta>
    <valor>8</valor>
  </respuesta>
  <respuesta>
    <valor>9</valor>
  </respuesta>
```

```
</respuesta>
<respuesta>
  <textorespuesta>10=Totalmente alegre</textorespuesta>
  <valor>10</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D20_Tristeza</id>
  <texto>¿Qué intensidad de TRISTEZA O ESTADO DE ÁNIMO BAJO sientes AHORA?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Nada triste</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <valor>7</valor>
  </respuesta>
  <respuesta>
    <valor>8</valor>
  </respuesta>
  <respuesta>
    <valor>9</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>10=Totalmente triste</textorespuesta>
    <valor>10</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D21_Ansiedad</id>
  <texto>¿Qué intensidad de ANSIEDAD O NERVIOSISMO sientes AHORA?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0=Nada
Ansioso/a</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <valor>4</valor>
  </respuesta>
```

```
<respuesta>
  <valor>5</valor>
</respuesta>
<respuesta>
  <valor>6</valor>
</respuesta>
<respuesta>
  <valor>7</valor>
</respuesta>
<respuesta>
  <valor>8</valor>
</respuesta>
<respuesta>
  <valor>9</valor>
</respuesta>
<respuesta>
  <textorespuesta>10=Totalmente
```

```
Ansioso/a</textorespuesta>
```

```
  <valor>10</valor>
```

```
</respuesta>
```

```
</pregunta>
```

```
<pregunta>
```

```
  <id>D41_Enfado</id>
```

```
  <texto>¿Qué intensidad de ENFADO sientes AHORA?</texto>
```

```
  <numeroRespuestas>11</numeroRespuestas>
```

```
  <tipo>Slider</tipo>
```

```
  <tiporespuesta>1</tiporespuesta>
```

```
  <respuesta>
```

```
    <textorespuesta>0=Nada enfadado/a</textorespuesta>
```

```
    <valor>0</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <valor>1</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <valor>2</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <valor>3</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <valor>4</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <valor>5</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <valor>6</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <valor>7</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <valor>8</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <valor>9</valor>
```

```
  </respuesta>
```

```
  <respuesta>
```

```
    <textorespuesta>10=Totalmente enfadado/a</textorespuesta>
```

```
    <valor>10</valor>
```

```
  </respuesta>
```

```
</pregunta>
```

```
<pregunta>
```

```
  <id>D29_InterferenciaTrabajo</id>
```

```
  <texto>¿Ha interferido HOY el dolor en tu capacidad para realizar tu TRABAJO HABITUAL o las TAREAS DOMÉSTICAS?</texto>
```

```
  <numeroRespuestas>11</numeroRespuestas>
```

```
  <tipo>Slider</tipo>
```

```
  <tiporespuesta>1</tiporespuesta>
```

```
  <respuesta>
```

Interferencia<textorespuesta>0=Ninguna

</textorespuesta>
<valor>0</valor>
</respuesta>
<respuesta>
<valor>1</valor>
</respuesta>
<respuesta>
<valor>2</valor>
</respuesta>
<respuesta>
<valor>3</valor>
</respuesta>
<respuesta>
<valor>4</valor>
</respuesta>
<respuesta>
<valor>5</valor>
</respuesta>
<respuesta>
<valor>6</valor>
</respuesta>
<respuesta>
<valor>7</valor>
</respuesta>
<respuesta>
<valor>8</valor>
</respuesta>
<respuesta>
<valor>9</valor>
</respuesta>
<respuesta>

Interferencia<textorespuesta>10=Máxima

</textorespuesta>
<valor>10</valor>
</respuesta>
</pregunta>
<pregunta>
<id>D30_InterferenciaOcio</id>
<texto>¿Ha interferido HOY el dolor en tus actividades de OCIO?</texto>
<numeroRespuestas>11</numeroRespuestas>
<tipo>Slider</tipo>
<tiporespuesta>1</tiporespuesta>
<respuesta>

Interferencia<textorespuesta>0=Ninguna

</textorespuesta>
<valor>0</valor>
</respuesta>
<respuesta>
<valor>1</valor>
</respuesta>
<respuesta>
<valor>2</valor>
</respuesta>
<respuesta>
<valor>3</valor>
</respuesta>
<respuesta>
<valor>4</valor>
</respuesta>
<respuesta>
<valor>5</valor>
</respuesta>
<respuesta>
<valor>6</valor>
</respuesta>
<respuesta>
<valor>7</valor>
</respuesta>
<respuesta>
<valor>8</valor>

```
</respuesta>
<respuesta>
  <valor>9</valor>
</respuesta>
<respuesta>
  <textorespuesta>10=Máxima
```

```
Interferencia</textorespuesta>
  <valor>10</valor>
</respuesta>
```

```
</pregunta>
```

```
<pregunta>
```

```
<id>D31_InterferenciaSocial</id>
```

```
<texto>¿Ha interferido HOY el dolor en tus RELACIONES CON OTRAS PERSONAS?</texto>
```

```
<numeroRespuestas>11</numeroRespuestas>
```

```
<tipo>Slider</tipo>
```

```
<tiporespuesta>1</tiporespuesta>
```

```
<respuesta>
```

```
<textorespuesta>0=Ninguna
```

```
Interferencia</textorespuesta>
```

```
<valor>0</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>1</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>2</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>3</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>4</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>5</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>6</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>7</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>8</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<valor>9</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<textorespuesta>10=Máxima
```

```
Interferencia</textorespuesta>
```

```
<valor>10</valor>
```

```
</respuesta>
```

```
</pregunta>
```

```
<pregunta>
```

```
<id>D32_Afrontamiento</id>
```

```
<texto>¿Qué ESTRATEGIA PARA AFRONTAR EL DOLOR has utilizado más HOY? Puedes marcar mas de una</texto>
```

```
<numeroRespuestas>8</numeroRespuestas>
```

```
<tipo>Check</tipo>
```

```
<tiporespuesta>1</tiporespuesta>
```

```
<respuesta>
```

```
<textorespuesta>Inactividad/descanso</textorespuesta>
```

```
<valor>1</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<textorespuesta>Relajación</textorespuesta>
```

```
<valor>2</valor>
```

```
</respuesta>
```

```
<respuesta>
```

```
<textorespuesta>Hablar con alguien</textorespuesta>
```

```
    <valor>3</valor>
  </respuesta>
</respuesta>
  <textorespuesta>Actividad física/Estiramientos</textorespuesta>
  <valor>4</valor>
</respuesta>
</respuesta>
  <textorespuesta>Darme instrucciones de ánimo y continuar con mis
  tareas</textorespuesta>
  <valor>5</valor>
</respuesta>
</respuesta>
  <textorespuesta>Hacer algo para sentir emociones más
  positivas</textorespuesta>
  <valor>6</valor>
</respuesta>
</respuesta>
  <textorespuesta>Ignorar el dolor y distraerme</textorespuesta>
  <valor>7</valor>
</respuesta>
</respuesta>
  <textorespuesta>Rezar para que el dolor mejore</textorespuesta>
  <valor>8</valor>
</respuesta>
</pregunta>
```

```
<pregunta>
  <id>D34_Magnificación</id>
  <texto>Indica tu grado de acuerdo con esta afirmación: Temo que el dolor
  empeore.</texto>
  <numeroRespuestas>5</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Nada de acuerdo </textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Poco de acuerdo </textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Ni de acuerdo, ni en desacuerdo </textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Bastante de acuerdo </textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Totalmente de acuerdo </textorespuesta>
    <valor>5</valor>
  </respuesta>
</pregunta>
```

```
<pregunta>
  <id>D35_Rumiacion</id>
  <texto>Indica tu grado de acuerdo con esta afirmación: Hoy no he podido apartar
  el dolor de mi mente</texto>
  <numeroRespuestas>5</numeroRespuestas>
  <tipo>Radio</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Nada de acuerdo </textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Poco de acuerdo </textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Ni de acuerdo, ni en desacuerdo </textorespuesta>
    <valor>3</valor>
```

```
</respuesta>
<respuesta>
  <textorespuesta>Bastante de acuerdo </textorespuesta>
  <valor>4</valor>
</respuesta>
<respuesta>
  <textorespuesta>Totalmente de acuerdo </textorespuesta>
  <valor>5</valor>
</respuesta>
</pregunta>
<pregunta>
  <id>D36_Actividad</id>
  <texto>¿Qué PORCENTAJE del día has estado activo/a HOY?</texto>
  <numeroRespuestas>11</numeroRespuestas>
  <tipo>Slider</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>0%=Nada activo</textorespuesta>
    <valor>0</valor>
  </respuesta>
  <respuesta>
    <valor>10</valor>
  </respuesta>
  <respuesta>
    <valor>20</valor>
  </respuesta>
  <respuesta>
    <valor>30</valor>
  </respuesta>
  <respuesta>
    <valor>40</valor>
  </respuesta>
  <respuesta>
    <valor>50</valor>
  </respuesta>
  <respuesta>
    <valor>60</valor>
  </respuesta>
  <respuesta>
    <valor>70</valor>
  </respuesta>
  <respuesta>
    <valor>80</valor>
  </respuesta>
  <respuesta>
    <valor>90</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>100%=Totalmente activo</textorespuesta>
    <valor>100</valor>
  </respuesta>
</pregunta>
<pregunta>
  <id>D37_Area</id>
  <texto>¿A qué ÁREA te has dedicado más HOY? Puedes marcar mas de una</texto>
  <numeroRespuestas>7</numeroRespuestas>
  <tipo>Check</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Trabajo</textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Familia</textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Pareja</textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
```

```
<textorespuesta>Amigos</textorespuesta>
<valor>4</valor>
</respuesta>
<respuesta>
<textorespuesta>Ocio</textorespuesta>
<valor>5</valor>
</respuesta>
<respuesta>
<textorespuesta>Actividad física</textorespuesta>
<valor>6</valor>
</respuesta>
<respuesta>
<textorespuesta>Otra</textorespuesta>
<valor>7</valor>
</respuesta>
</pregunta>
<pregunta>
<id>D38_Rescate</id>
<texto>¿Has tomado HOY algún medicamento de rescate, es decir, un medicamento
que tomas solo cuando tienes picos de dolor?</texto>
<numeroRespuestas>2</numeroRespuestas>
<tipo>Radio</tipo>
<tiporespuesta>1</tiporespuesta>
<respuesta>
<textorespuesta>Sí</textorespuesta>
<valor>1</valor>
</respuesta>
<respuesta>
<textorespuesta>No</textorespuesta>
<valor>0</valor>
</respuesta>
</pregunta>
<pregunta>
<id>D39_Secundarios1</id>
<texto>¿Has tenido HOY alguno de estos síntomas? Puedes marcar mas de uno:</texto>
<numeroRespuestas>8</numeroRespuestas>
<tipo>Check</tipo>
<tiporespuesta>1</tiporespuesta>
<respuesta>
<textorespuesta>Náuseas/vómitos </textorespuesta>
<valor>1</valor>
</respuesta>
<respuesta>
<textorespuesta>Taquicardia</textorespuesta>
<valor>2</valor>
</respuesta>
<respuesta>
<textorespuesta>Estreñimiento</textorespuesta>
<valor>3</valor>
</respuesta>
<respuesta>
<textorespuesta>Somnolencia/sedación </textorespuesta>
<valor>4</valor>
</respuesta>
<respuesta>
<textorespuesta>Visión borrosa </textorespuesta>
<valor>5</valor>
</respuesta>
<respuesta>
<textorespuesta>Boca seca </textorespuesta>
<valor>6</valor>
</respuesta>
<respuesta>
<textorespuesta>Dolor de cabeza </textorespuesta>
<valor>7</valor>
</respuesta>
<respuesta>
<textorespuesta>Ninguno de los anteriores</textorespuesta>
<valor>8</valor>
</respuesta>
</pregunta>
```

```
<pregunta>
  <id>D40_Secundarios2</id>
  <texto>Para finalizar, ¿has tenido HOY alguno de estos otros síntomas? Puedes
  marcar mas de uno</texto>
  <numeroRespuestas>7</numeroRespuestas>
  <tipo>Check</tipo>
  <tiporespuesta>1</tiporespuesta>
  <respuesta>
    <textorespuesta>Mareos </textorespuesta>
    <valor>1</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Picor</textorespuesta>
    <valor>2</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Diarrea</textorespuesta>
    <valor>3</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Inestabilidad al caminar </textorespuesta>
    <valor>4</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Sudoración excesiva </textorespuesta>
    <valor>5</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Otro síntoma distinto </textorespuesta>
    <valor>6</valor>
  </respuesta>
  <respuesta>
    <textorespuesta>Ninguno de los anteriores </textorespuesta>
    <valor>7</valor>
  </respuesta>
</pregunta>
</lanzar>
</config>
```