

**Trabajo Fin de ciclo**

|  |
| --- |
| **ADOPTALES** |

**CICLO FORMATIVO DE GRADO SUPERIOR**

**DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

FECHA



del Alumno/a:

Ruben Palomo Yuste

Director/a TFM:

Carlos Rufiángel García

Contenido:

1. RESUMEN DEL PROYECTO
2. AGRADECIMIENTOS
3. ÍNDICE DE CAPÍTULOS
4. ÍNDICE DE TABLAS
5. ÍNDICE DE FIGURAS
6. INTRODUCCIÓN
7. DESCRIPCIÓN DEL PROBLEMA
8. ESTADO DE LA CUESTIÓN
9. OBJETIVO GENERAL
10. OBJETIVOS ESPECÍFICOS
11. SOLUCIÓN PROPUESTA
12. METODOLOGÍA
13. ESPECIFICACIÓN DE REQUISITOS DE USUARIO.
14. CASOS DE USO.
15. DISEÑO CONCEPTUAL.
16. DISEÑO DETALLADO.
17. REQUISITOS DEL SISTEMA.
18. HERRAMIENTAS EMPLEADAS EN LA IMPLEMENTACIÓN.
19. DETALLES DE IMPLEMENTACIÓN.
20. ASEGURAMIENTO DE LA CALIDAD.
21. EXPERIMENTACIÓN/ PRUEBAS/ VALIDACIÓN.
22. EVALUACIÓN/ DISCUSIÓN.
23. CONCLUSIONES.
24. TRABAJOS FUTUROS
25. GUÍA DE SOLUCIÓN DE ERRORES.
26. GLOSARIO DE TÉRMINOS.
27. BIBLIOGRAFÍA.
28. APENDICES
29. RESUMEN DEL PROYECTO.

Mi proyecto trata de una aplicación para adoptar mascotas de refugios de animales o personas que ya no se pueden permitir tener una mascota en casa y también contiene información sobre refugios de animales.

My project is about an application to adopt pets from animal shelters or people who can no longer afford to have a pet at home and contains information about animal shelters.

1. AGRADECIMIENTOS

Mis agradecimientos para empezar van para mis padres por pagarme el curso y darme esta gran oportunidad, segundo a mi novia por darme la gran idea de esta app y tercero a todos mis profesores por todo lo que me han enseñado, que no es poco.

1. ÍNDICE DE CAPÍTULOS

[1. RESUMEN DEL PROYECTO. 3](#_Toc514168703)

[2. AGRADECIMIENTOS 3](#_Toc514168704)

[3. ÍNDICE DE CAPÍTULOS 4](#_Toc514168705)

[4. ÍNDICE DE TABLAS 6](#_Toc514168706)

[6. INTRODUCCIÓN 6](#_Toc514168707)

[7. DESCRIPCIÓN DEL PROBLEMA 6](#_Toc514168708)

[8. ESTADO DE LA CUESTIÓN 6](#_Toc514168709)

[9. OBJETIVO GENERAL 7](#_Toc514168710)

[10. OBJETIVOS ESPECÍFICOS 7](#_Toc514168711)

[11. SOLUCIÓN PROPUESTA 8](#_Toc514168712)

[12. METODOLOGÍA 8](#_Toc514168713)

[13. ESPECIFICACIÓN DE REQUISITOS DE USUARIO. 9](#_Toc514168714)

[14. CASOS DE USO. 10](#_Toc514168715)

[15. DISEÑO CONCEPTUAL. 11](#_Toc514168716)

[16. DISEÑO DETALLADO. 13](#_Toc514168717)

[1. Diagrama de Gantt: 13](#_Toc514168718)

[2. Base de datos: 15](#_Toc514168719)

[3. Diagrama de clases: 16](#_Toc514168720)

[17. REQUISITOS DEL SISTEMA. 17](#_Toc514168721)

[18. HERRAMIENTAS EMPLEADAS EN LA IMPLEMENTACIÓN. 17](#_Toc514168722)

[19. DETALLES DE IMPLEMENTACIÓN. 18](#_Toc514168723)

[20. IMÁGENES UTILIZADAS. 191](#_Toc514168724)

[21. ASEGURAMIENTO DE LA CALIDAD. 191](#_Toc514168725)

[22. EXPERIMENTACIÓN/ PRUEBAS/ VALIDACIÓN. 191](#_Toc514168726)

[23. EVALUACIÓN/ DISCUSIÓN. 191](#_Toc514168727)

[24. CONCLUSIONES. 191](#_Toc514168728)

[25. TRABAJOS FUTUROS 191](#_Toc514168729)

[26. GUÍA DE SOLUCIÓN DE ERRORES. 191](#_Toc514168730)

[27. GLOSARIO DE TÉRMINOS. 191](#_Toc514168731)

[28. BIBLIOGRAFÍA. 191](#_Toc514168732)

[29. APENDICES 191](#_Toc514168733)

1. ÍNDICE DE TABLAS
2. ÍNDICE DE FIGURAS
3. INTRODUCCIÓN

Esta idea viene por la cuestión de que hoy en día hay que adoptar animales, no comprarlos y quiero ayudar a solucionar ese problema desarrollando una aplicación que permita ponerse en contacto con personas que estén dando a adoptar animales desde mi aplicación y que estos encuentren un hogar rápido.

Se va desarrollar en Android Studio, con una base de datos externa (MySql), otra interna (SQLite), y un registro para guardar datos de login.

1. DESCRIPCIÓN DEL PROBLEMA

Hoy en día hay un grave problema en la sociedad a la hora de adopción de animales ya que a estas fechas todavía se compran animales y eso está muy mal, lo que quiero es ayudar a acabar con esto y que solo se adopten animales y mi aplicación va a ser para ayudar un poco más a resolver este problema y voy a hacer más fácil contactar con personas que pongan animales en adopción.

1. ESTADO DE LA CUESTIÓN

En este momento no hay muchas aplicaciones que ayuden a adoptar animales y lo que más veo son anuncios o por redes sociales, y alguna página web, pero en cuanto a app de Android no hay muchas hoy en día todo el mundo utiliza aplicaciones móviles y esto puede ayudar a muchos animales a encontrar un hogar gracias a mi aplicación.

1. OBJETIVO GENERAL

OG1. Va a ayudar a los animales a encontrar un hogar.

OG2. Va a ayudar a la persona que adoptan animales a buscarlos.

OG3. Va a ayudar a personas que den en adopción a animales ya que puede encontrar un hogar nuevo rápido.

OG4. Se va a ofrecer información de refugios de animales.

OG5. Base de datos donde se guarde la información.

1. OBJETIVOS ESPECÍFICOS

OG1. Va a ayudar a los animales a encontrar un hogar.

OE1.1. Subir los animales con su información.

OE1.2. Mostrar los animales según la fecha de subida.

OE1.3. Mostrar la forma de contacto para los animales.

OE1.4. Poder llamar al usuario vendedor.

OG2. Va a ayudar a la persona que adoptan animales a buscarlos.

OE2.1. Cargar todos los animales puestos en adopción.

OE2.2. Tener un filtro para hacer búsquedas más específicas.

OE2.3. Poder ver la información del animal y su forma de contacto.

OG3. Va a ayudar a personas que den en adopción a animales ya que pueden

encontrar un hogar nuevo rápido.

OE3.1. Subir animales. (Si te registras)

OE3.2. Comprobar que los datos son correctos. (Si te registras)

OE3.3 Borrar animales. (Si te registras)

OG4. Se va a ofrecer información de refugios de animales.

OE4.1. Ver información de refugios de animales.

OE4.2. Poder llamar al refugio.

OE4.3. Poder ver la localización del refugio.

OG5. Base de datos donde se guarde la información.

OE5.1. Base de datos MySQL, externa.

OE5.2. Web servicie para la conexión con la Base de Datos.

OE5.3. Base de datos SQLite, interna, donde se va a guardar los datos del MySQL, para solo acceder una vez.

OE5.4. Un registro donde guardo los datos del login y la versión de las Bases de Datos

1. SOLUCIÓN PROPUESTA

En base a las especificaciones los más necesario es hacer una aplicación en Java y XML con Android Studio que contengan interfaces y donde nada más iniciar el programa se vean los animales, luego para subir animales te tienes que registrar o lograr en la app y ahí podrás subir y borrar tus animales. Tendremos un apartado en el menú donde estará la información de los refugios de animales.

Mi solución es esta ya que es la más fácil de entender por los usuarios y es más sencilla de implementar en una aplicación.

1. METODOLOGÍA

Voy a utilizar una metodología en cascada ya que es la que mejor me manejo y su función es la que más me gusta utilizar, funciona con ver un problema que hay hoy en día, ver los requisitos que necesitamos e solucionarlo desarrollando una aplicación para este problema y luego realizar pruebas de funcionamiento mientras se van realizando funcionamientos.

1. ESPECIFICACIÓN DE REQUISITOS DE USUARIO.

**USUARIO ANONIMO.**

R1. Ver todos los animales en adopción.

T1.1. Ver todos los datos de los animales.

T1.2. Ver la información del animal.

T1.3. Llamar al contacto del animal.

R2. Ver los refugios de animales.

T2.1. Ver la información de los refugios.

T2.2. Llamar al refugio.

T2.3. Ver la localización del refugio.

R3. Registro de usuario.

T3.1. Pedir datos en el formulario.

T3.2. Comprobar que los datos son correctos.

T3.3. Meter los datos del usuario en la base de datos.

T3.4. Enviar correo de registro.

R4. Login de usuarios.

T4.1. Pedir datos del login.

T4.2. Comprobar que los datos son correctos.

T4.3. Comprobar los datos del usuario con la base de datos.

R5. Información y ayuda de la aplicación.

T5.1. Mostrar información sobre el desarrollador.

T5.2. Una forma de contacto con el desarrollador.

T5.3. Mostrar una ayuda del uso de la aplicación.

**USUARIO REGISTRADO.**

R1. Ver todos los animales en adopción.

T1.1. Ver todos los datos de los animales.

T1.2. Ver la información del animal.

T1.3. Llamar al contacto del animal.

R2. Ver los refugios de animales.

T2.1. Ver la información de los refugios.

T2.2. Llamar al refugio.

T2.3. Ver la localización del refugio.

R3. Registro de usuario.

T3.1. Pedir datos en el formulario.

T3.2. Comprobar que los datos son correctos.

T3.3. Meter los datos del usuario en la base de datos.

T3.4. Enviar correo de registro.

R4. Login de usuarios.

T4.1. Pedir datos del login.

T4.2. Comprobar que los datos son correctos.

T4.3. Comprobar los datos del usuario con la base de datos.

R5. Subida de animales.

T5.1. Pedir datos del animal.

T5.2. Comprobar que los datos son correctos.

T5.3. Meter los datos del animal en la base de datos.

R6. Borrar animales.

T6.1. Pedir el nombre del animal a borrar.

T6.2. Comprobar que el dato es correcto.

T6.2. Borrar el animal de la base de datos.

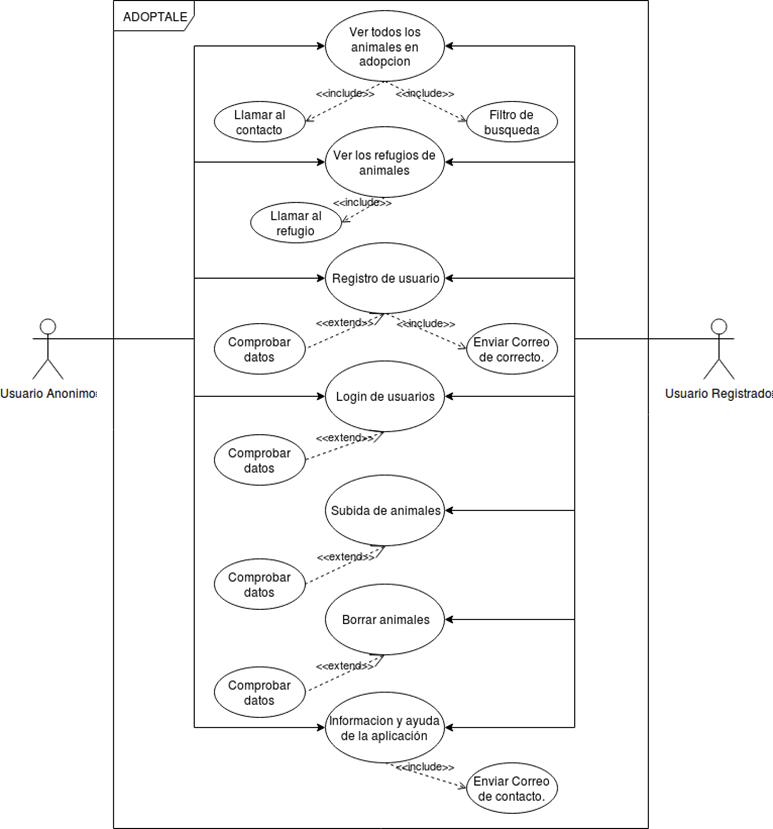
R7. Información y ayuda de la aplicación.

T7.1. Mostrar información sobre el desarrollador.

T7.2. Una forma de contacto con el desarrollador.

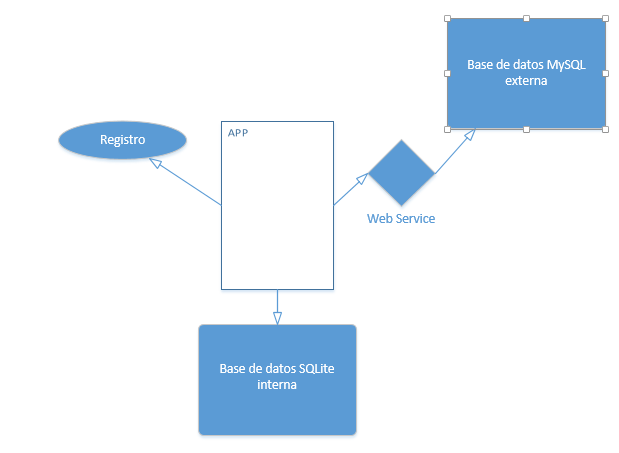
T7.3. Mostrar una ayuda del uso de la aplicación.

1. CASOS DE USO.

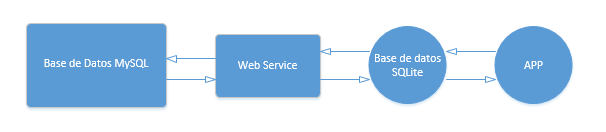


1. DISEÑO CONCEPTUAL.

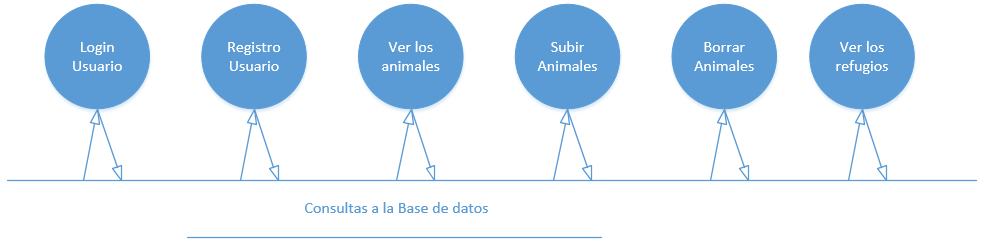
N1.



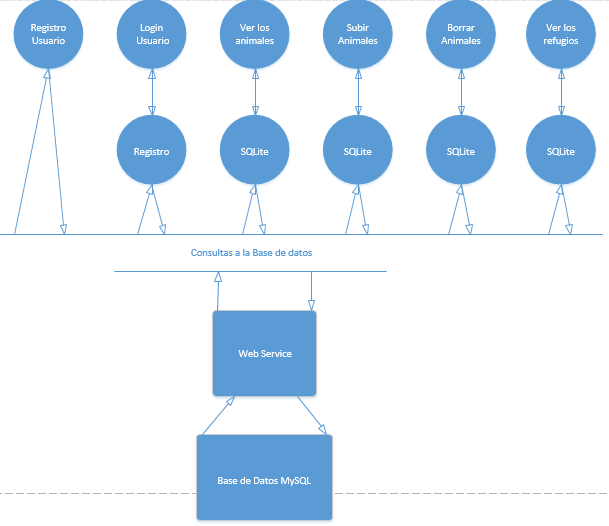
N2.



N3.

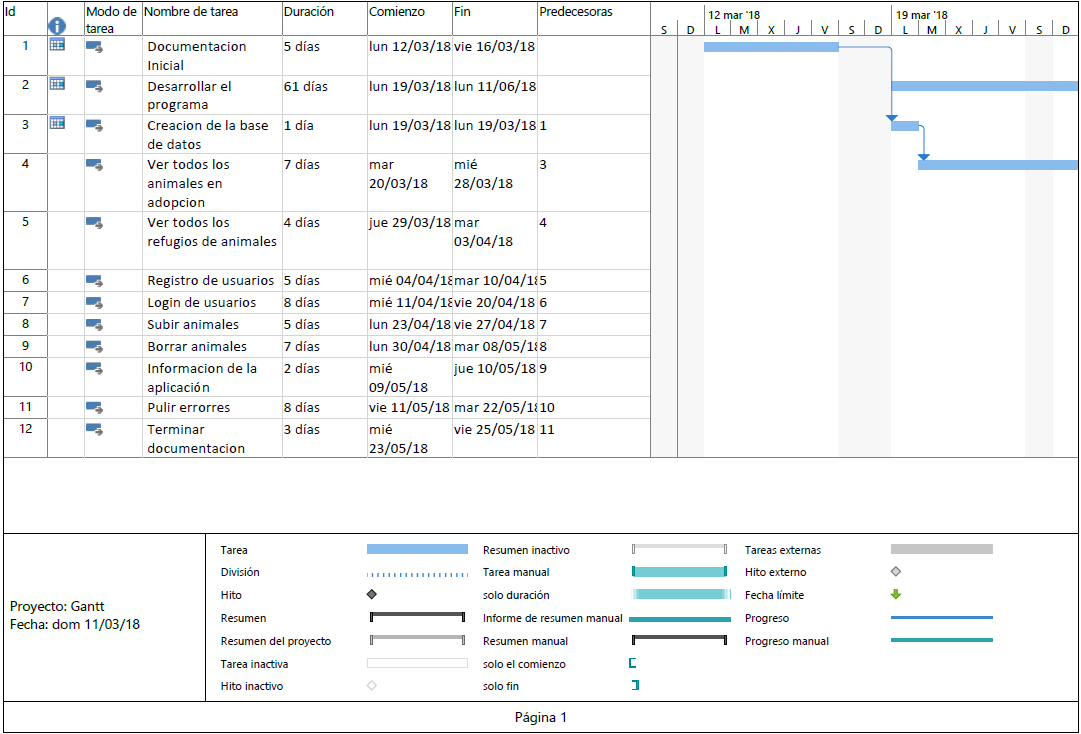


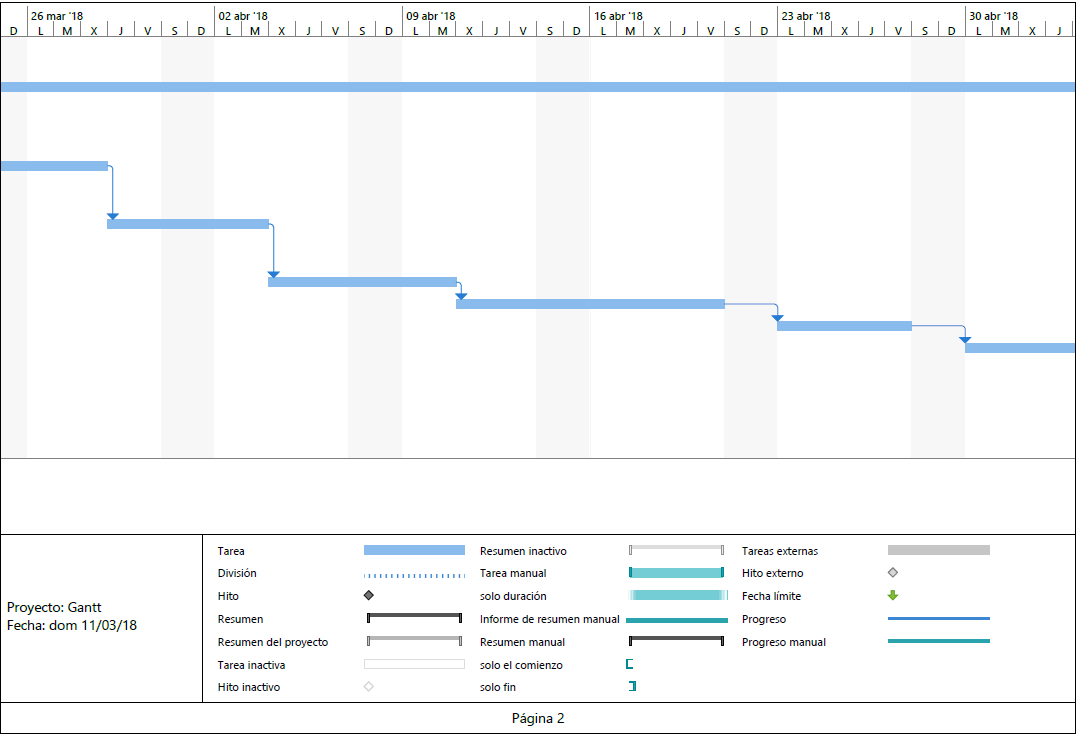
N4.

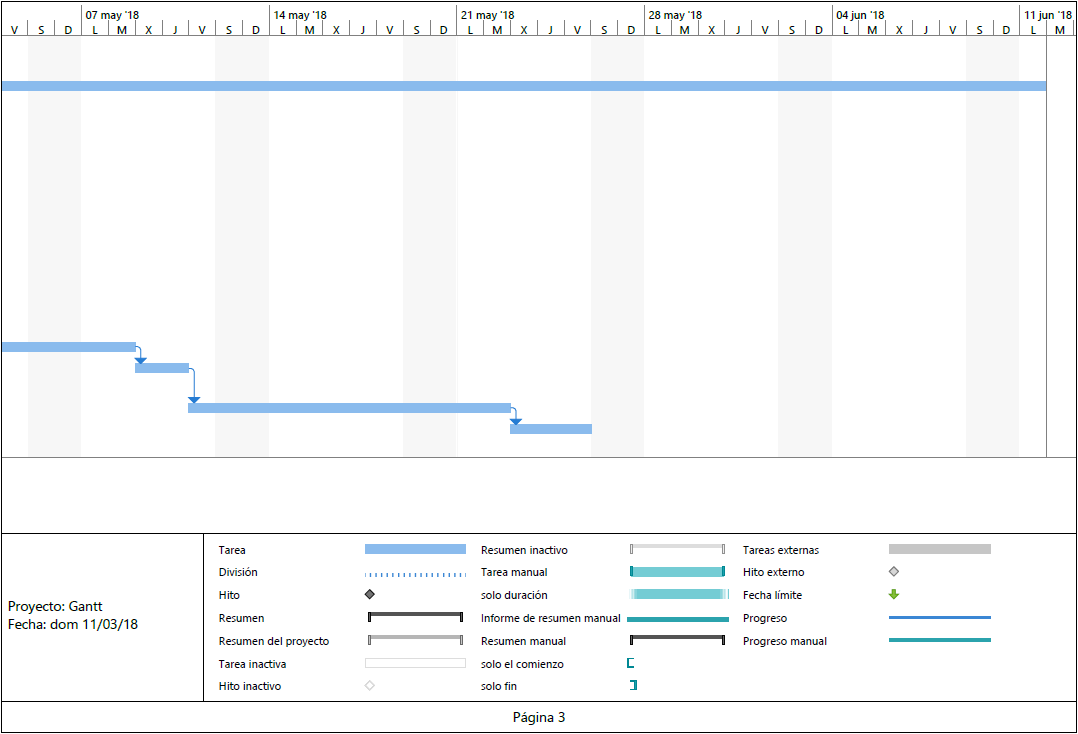


1. DISEÑO DETALLADO.
2. Diagrama de Gantt:

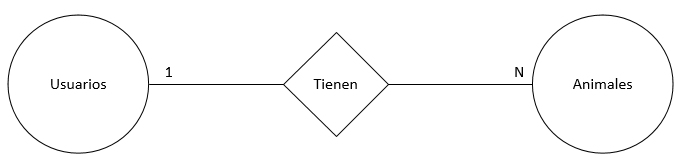
[Pulse aquí para ver el diagrama de Gantt.](Gantt.pdf)

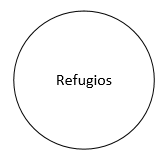


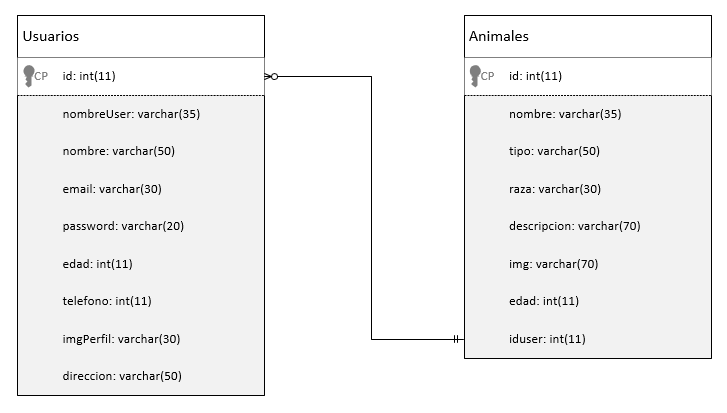




1. Base de datos:

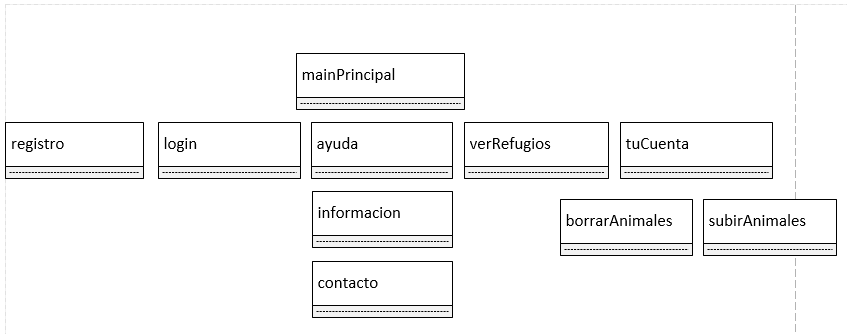








1. Diagrama de clases:



1. REQUISITOS DEL SISTEMA.

RS1. Móvil Android.

RS2. API 19 o mayor.

RS3. 30 MB de espacio.

RS4. 1 GB de RAM.

1. HERRAMIENTAS EMPLEADAS EN LA IMPLEMENTACIÓN.

H1. Android Studio.

H2. MySQL, PhpMyAdmin.

H3. Photoshop.

H4. Microsoft Office.

H5. Libre Office.

H6. Draw.io ([www.draw.io](http://www.draw.io))

H7. Adobe Reader.

1. DETALLES DE IMPLEMENTACIÓN.

DI1. AdaptadorAnimales

package com.adoptales.rubenpalomo.adoptales;  
  
import android.app.Activity;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.graphics.Typeface;  
import android.os.AsyncTask;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.squareup.picasso.Picasso;  
  
import java.io.IOException;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
*/\*\*  
 \* Created by Ruben Palomo on 13/12/2017.  
 \*/*class AdaptadorAnimales extends ArrayAdapter<String> {  
  
 // Atributos del objeto  
 private final Activity context;  
 private final String[] nombreMeme;  
 private final String[] imgAnimal;  
 Typeface texto;  
  
  
 */\*\*  
 \* Constructor  
 \** ***@param*** *context  
 \** ***@param*** *nombreMeme  
 \*/* public AdaptadorAnimales(Activity context, String[] nombreMeme, String[] imgAnimal, Typeface texto) {  
 super(context, R.layout.*listaanimales*, nombreMeme);  
 // *TODO Auto-generated constructor stub* // Actualizamos el contenido  
 this.context=context;  
 this.nombreMeme=nombreMeme;  
 this.imgAnimal=imgAnimal;  
 this.texto = texto;  
 }  
  
 */\*\*  
 \* Ponemos la vista  
 \** ***@param*** *posicion  
 \** ***@param*** *view  
 \** ***@param*** *parent  
 \** ***@return*** *\*/* public View getView(int posicion,View view, ViewGroup parent){  
  
 // Creamos una variable de tipo LayoutInflater y lo igualamos metodo de la clase Activity  
 LayoutInflater inflater = LayoutInflater.*from*(context);  
 // Hacemos una vista en el listView  
 View rowView=inflater.inflate(R.layout.*listaanimales*,null);  
  
 // Inizializamos los widget  
 TextView txtTitle = rowView.findViewById(R.id.*nombreLista*);  
 ImageView imageView = rowView.findViewById(R.id.*img*);  
  
 // Ponemos el texto en el textView  
 txtTitle.setText(nombreMeme[posicion]);  
 txtTitle.setTypeface(texto);  
  
 // Cogemos y ponemos la imagen del servidor  
 Picasso.*with*(context)  
 .load("https://www.losnaranjosdam.online/2dam/ruben/adoptales/fotos/"+ imgAnimal[posicion])  
 .error(R.mipmap.*ic\_launcher*)  
 .into(imageView);  
  
 // Retornamos la vista  
 return rowView;  
 }  
}

DI2. AdaptadorRefugio.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.app.Activity;  
import android.graphics.Typeface;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.squareup.picasso.Picasso;  
  
*/\*\*  
 \* Created by Ruben Palomo on 13/12/2017.  
 \*/*class AdaptadorRefugio extends ArrayAdapter<String> {  
  
 // Atributos del objeto  
 private final Activity context;  
 private final String[] nombreRefugio;  
 private final String[] imgRefugio;  
 Typeface texto;  
  
 */\*\*  
 \* Constructor  
 \** ***@param*** *context  
 \** ***@param*** *nombreRefugio  
 \** ***@param*** *imgRefugio  
 \*/* public AdaptadorRefugio(Activity context, String[] nombreRefugio, String[] imgRefugio, Typeface texto) {  
 super(context, R.layout.*listarefugio*, nombreRefugio);  
 // *TODO Auto-generated constructor stub* // Actualizamos el contenido  
 this.context=context;  
 this.nombreRefugio=nombreRefugio;  
 this.imgRefugio=imgRefugio;  
 this.texto = texto;  
 }  
  
 */\*\*  
 \* Ponemos la vista  
 \** ***@param*** *posicion  
 \** ***@param*** *view  
 \** ***@param*** *parent  
 \** ***@return*** *\*/* public View getView(int posicion,View view, ViewGroup parent){  
  
 // Creamos una variable de tipo LayoutInflater y lo igualamos metodo de la clase Activity  
 LayoutInflater inflater=context.getLayoutInflater();  
 // Hacemos una vista en el listView  
 View rowView=inflater.inflate(R.layout.*listarefugio*,null,true);  
  
 // Inizializamos los widget  
 TextView txtTitle = rowView.findViewById(R.id.*nombreListaRefugios*);  
 ImageView imageView = rowView.findViewById(R.id.*img*);  
  
 // Ponemos el texto en el textView  
 txtTitle.setText(nombreRefugio[posicion]);  
 txtTitle.setTypeface(texto);  
  
 // Cogemos y ponemos la imagen del servidor  
 Picasso.*with*(context)  
 .load("https://www.losnaranjosdam.online/2dam/ruben/adoptales/fotosRefugio/"+ imgRefugio[posicion])  
 .error(R.mipmap.*ic\_launcher*)  
 .into(imageView);  
  
 // Retornamos la vista  
 return rowView;  
 }  
}

DI3. AdaptadorTusAnimales.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.app.Activity;  
import android.graphics.Typeface;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.squareup.picasso.Picasso;  
  
*/\*\*  
 \* Created by Ruben Palomo on 13/12/2017.  
 \*/*class AdaptadorTusAnimales extends ArrayAdapter<String> {  
  
 // Atributos del objeto  
 private final Activity context;  
 private final String[] nombreMeme;  
 private final String[] imgAnimal;  
 Typeface texto;  
  
 */\*\*  
 \* Constructor  
 \** ***@param*** *context  
 \** ***@param*** *nombreMeme  
 \*/* public AdaptadorTusAnimales(Activity context, String[] nombreMeme, String[] imgAnimal, Typeface texto) {  
 super(context, R.layout.*listaanimales*, nombreMeme);  
 // *TODO Auto-generated constructor stub* // Actualizamos el contenido  
 this.context=context;  
 this.nombreMeme=nombreMeme;  
 this.imgAnimal=imgAnimal;  
 this.texto = texto;  
 }  
  
 */\*\*  
 \* Ponemos la vista  
 \** ***@param*** *posicion  
 \** ***@param*** *view  
 \** ***@param*** *parent  
 \** ***@return*** *\*/* public View getView(int posicion,View view, ViewGroup parent){  
  
 // Creamos una variable de tipo LayoutInflater y lo igualamos metodo de la clase Activity  
 LayoutInflater inflater = LayoutInflater.*from*(context);  
 // Hacemos una vista en el listView  
 View rowView=inflater.inflate(R.layout.*listaanimales*,null);  
  
 // Inizializamos los widget  
 TextView txtTitle = rowView.findViewById(R.id.*nombreLista*);  
 ImageView imageView = rowView.findViewById(R.id.*img*);  
  
 // Ponemos el texto en el textView  
 txtTitle.setText(nombreMeme[posicion]);  
 txtTitle.setTypeface(texto);  
  
 Picasso.*with*(context)  
 .load("https://www.losnaranjosdam.online/2dam/ruben/adoptales/fotos/"+ imgAnimal[posicion])  
 .error(R.mipmap.*ic\_launcher*)  
 .into(imageView);  
  
 // Retornamos la vista  
 return rowView;  
 }  
}

DI4. Animal.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.annotation.SuppressLint;  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.content.pm.PackageManager;  
import android.graphics.Typeface;  
import android.net.Uri;  
import android.os.AsyncTask;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.StrictMode;  
import android.support.annotation.NonNull;  
import android.support.design.widget.BottomNavigationView;  
import android.support.design.widget.NavigationView;  
import android.support.v4.app.ActivityCompat;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.util.Log;  
import android.view.Display;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.squareup.picasso.Picasso;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Properties;  
  
import javax.mail.Authenticator;  
import javax.mail.Message;  
import javax.mail.MessagingException;  
import javax.mail.PasswordAuthentication;  
import javax.mail.Session;  
import javax.mail.Transport;  
import javax.mail.internet.AddressException;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class animal extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // TextView del refugio  
 TextView txtNombreAnimal, txtEdadAnimal, txtEmailAnimal, txtEmail, txtMensaje, txtDescripcionAnimal, txtTipoAnimal  
 , txtNumeroAnimal, txtChip, txtVacuna, txtGenero, txtInformacion, txtInfoAnimal;  
  
 // Button del refugio  
 Button btnEnviarAnimal;  
  
 LinearLayout ly1, ly2, ly3, ly4;  
  
 // ImageView del refugio  
 ImageView imgAnimal;  
  
 // EditText de contacto  
 EditText edtMensaje, edtEmail;  
  
 String id;  
 String nombre;  
 String raza;  
 String tipo;  
 String descripcion;  
 String imganimal;  
 String edad;  
 String iduser;  
 String chip;  
 String vacuna;  
 String genero;  
  
 private static final String *TAG\_USUARIOSINFO* = "usuario";  
 private static final String *TAG\_TELEFONO* = "telefono";  
 private static final String *TAG\_EMAIL* = "email";  
 private static final String *TAG\_DIRECCION* = "direccion";  
  
 // Declaro variables  
 String correo;  
 String password;  
 // Creo una sesion  
 Session session;  
  
 String email;  
 String telefono;  
 String direccion;  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 private static String *url*;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_animal*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.*navigation*);  
 navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
  
 Display display = getWindowManager().getDefaultDisplay();  
 int width = display.getWidth() / 4; // obsoleto (deprecated)  
  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 ly1 = findViewById(R.id.*ly1*);  
 ly2 = findViewById(R.id.*ly2*);  
 ly3 = findViewById(R.id.*ly3*);  
 ly4 = findViewById(R.id.*ly4*);  
  
 ly1.getLayoutParams().width=width;  
 ly2.getLayoutParams().width=width;  
 ly3.getLayoutParams().width=width;  
 ly4.getLayoutParams().width=width;  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 // Inicializamos los widgets  
 txtNombreAnimal = findViewById(R.id.*txtNombreAnimal*);  
 txtEdadAnimal = findViewById(R.id.*txtEdadAnimal*);  
 txtTipoAnimal = findViewById(R.id.*txtTipoAnimal*);  
 txtEmailAnimal = findViewById(R.id.*txtEmailAnimal*);  
 txtNumeroAnimal = findViewById(R.id.*txtNumeroAnimal*);  
 txtEmail = findViewById(R.id.*txtEmail*);  
 txtMensaje = findViewById(R.id.*txtMensaje*);  
 txtDescripcionAnimal = findViewById(R.id.*txtDescripcionAnimal*);  
 txtChip = findViewById(R.id.*txtChip*);  
 txtVacuna = findViewById(R.id.*txtVacuna*);  
 txtGenero = findViewById(R.id.*txtGenero*);  
 txtInformacion = findViewById(R.id.*txtInformacion*);  
 txtInfoAnimal = findViewById(R.id.*txtInfoAnimal*);  
 edtMensaje = findViewById(R.id.*edtMensaje*);  
 edtEmail = findViewById(R.id.*edtEmail*);  
 btnEnviarAnimal = findViewById(R.id.*btnEnviarAnimal*);  
 imgAnimal = findViewById(R.id.*imgAnimal*);  
  
 // Ponemos las fuentes  
 txtNombreAnimal.setTypeface(enunciado);  
 txtEdadAnimal.setTypeface(texto);  
 txtEmail.setTypeface(texto);  
 txtMensaje.setTypeface(texto);  
 txtInformacion.setTypeface(texto);  
 txtInfoAnimal.setTypeface(texto);  
 txtVacuna.setTypeface(texto);  
 txtGenero.setTypeface(texto);  
 txtChip.setTypeface(texto);  
 btnEnviarAnimal.setTypeface(texto);  
  
 // Cogemos los extras que nos pasan en la anterio activity  
 Bundle extras = getIntent().getExtras();  
  
 // Si los extras no son nulos  
 if (extras != null) {  
 // Guardo los datos que tengan como clave ano  
 id = extras.getString("id");  
 nombre = extras.getString("nombre");  
 raza = extras.getString("raza");  
 tipo = extras.getString("tipo");  
 descripcion = extras.getString("descripcion");  
 imganimal = extras.getString("imgAnimal");  
 edad = extras.getString("edad");  
 iduser = extras.getString("iduser");  
 chip = extras.getString("chip");  
 vacuna = extras.getString("vacuna");  
 genero = extras.getString("genero");  
 } else {  
 // Se queda vacio  
 id = "";  
 nombre = "";  
 raza = "";  
 tipo = "";  
 descripcion = "";  
 imganimal = "";  
 edad = "";  
 iduser = "";  
 }  
  
 // Ponemos las fuentes  
 txtNombreAnimal.setText(nombre);  
 txtTipoAnimal.setText("-"+tipo+", "+ raza);  
 txtEdadAnimal.setText(edad);  
 txtDescripcionAnimal.setText("-"+descripcion);  
 txtChip.setText(chip);  
 txtVacuna.setText(vacuna);  
 txtGenero.setText(genero);  
  
 // Descargamos la imagen  
 Picasso.*with*(this)  
 .load("https://www.losnaranjosdam.online/2dam/ruben/adoptales/fotos/"+ imganimal)  
 .error(R.mipmap.*ic\_launcher*)  
 .into(imgAnimal);  
  
 // Ponemos la url  
 *url* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/datosAnimalUsuario.php?iduser="+ iduser;  
  
 // Ejecutamos el metodo GetDatosDueno  
 new GetDatosDueno().execute();  
 }  
  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/* private class GetDatosDueno extends AsyncTask<Void, Void, Void> {  
  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 // Creamos una instancia de la clase WebRequest  
 WebRequest webreq = new WebRequest();  
  
 // Obtenemos el json se los datos  
 String jsonStr = webreq.makeWebServiceCall(*url*, WebRequest.*GET*);  
  
 // Mostramos el json en el terminal  
 Log.*d*("Response: ", "> " + jsonStr);  
  
 // Ejecutamos el metodo que nos convierte el json  
 ParseJSON(jsonStr);  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 // Ponemos el email del animal.  
 txtEmailAnimal.setText("-"+ email);  
 txtNumeroAnimal.setText("-"+ telefono);  
  
 }  
 }  
  
 */\*\*  
 \* Metodo de analizar el json  
 \** ***@param*** *json  
 \*/* private void ParseJSON(String json) {  
 // Si es nulo  
 if (json != null) {  
 // Intentamos  
 try {  
 // Creamos un objeto JSONObject  
 JSONObject jsonObj = new JSONObject(json);  
  
 // Obtenemos el nodo del json pasado por parametro  
 JSONArray usuario = jsonObj.getJSONArray(*TAG\_USUARIOSINFO*);  
  
 // Recorremos todos los usuarios  
 for (int i = 0; i < usuario.length(); i++) {  
 // Obtenemos un elemento json  
 JSONObject c = usuario.getJSONObject(i);  
  
 // Obtenemos los datos de el elemento json  
 telefono = c.getString(*TAG\_TELEFONO*);  
 email = c.getString(*TAG\_EMAIL*);  
 direccion = c.getString(*TAG\_DIRECCION*);  
 }  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 } else {  
 // Mensaje de error por el terminal  
 Log.*e*("ServiceHandler", "Couldn't get any data from the url");  
 }  
 }  
  
 */\*\*  
 \* Comparamos el email  
 \** ***@return*** *\*/* public boolean compararEmail(){  
  
 // Declaramos variables  
 String email;  
 Boolean correcto;  
  
 // Inicializamos variables  
 email = edtEmail.getText().toString();  
 correcto = false;  
  
 // Si no esta vacio o es igual que un email  
 if(!email.isEmpty() &&  
 android.util.Patterns.*EMAIL\_ADDRESS*.matcher(email).matches()){  
 // Actualizamos variable  
 correcto = true;  
 }else{  
 // Mensaje de error  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el email." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean compararMensaje(){  
  
 // Declaramos variables  
 String mensaje;  
 Boolean correcto;  
  
 // Inicializamos variables  
 mensaje = edtMensaje.getText().toString();  
 correcto = false;  
  
 // Si no esta vacio  
 if(!mensaje.isEmpty()){  
 // Actualizamos variable  
 correcto = true;  
 }else{  
 // Mensaje de error  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el mensaje." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 */\*\*  
 \* Enviamos el correo al dueño  
 \** ***@param*** *v  
 \*/* public void enviarCorreo(View v){  
 // Si los dos metodos son correctos  
 if(compararEmail() && compararMensaje()){  
 // Pongo email y contraseña  
 correo = "adoptalescontacto@gmail.com";  
 password = "Ru83nP1l0m03105";  
  
 // Ponemos las politica  
 StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();  
 StrictMode.*setThreadPolicy*(policy);  
 // Creamos unas propiedades  
 Properties properties = new Properties();  
 // ponemos la conexion  
 properties.put("mail.smtp.auth", "true");  
 properties.put("mail.smtp.starttls.enable", "true");  
 properties.put("mail.smtp.host", "smtp.gmail.com");  
 properties.put("mail.smtp.port", "587");  
  
 // Intentamos  
 try {  
 // Inicializamos la sesion  
 session = Session.*getDefaultInstance*(properties, new Authenticator() {  
 @Override  
 // Autenticacion  
 protected PasswordAuthentication getPasswordAuthentication() {  
 return new PasswordAuthentication(correo, password);  
 }  
 });  
  
 // Si la sesion es erronea  
 if (session != null) {  
 // Iniciamos sesion  
 Message message = new MimeMessage(session);  
  
 // Quien lo envia  
 message.setFrom(new InternetAddress(correo));  
  
 // Titulo  
 message.setSubject("Mensaje de contacto");  
  
 // Quien lo recibe  
 message.setRecipients(Message.RecipientType.*TO*, InternetAddress.*parse*(email));  
  
 // Mensaje y su forma  
 message.setContent("Buenas, con email " + edtEmail.getText().toString() +  
 "<br>Te quiero decir: "  
 + edtMensaje.getText().toString(), "text/html; charset=utf-8");  
  
 // Enviamos el mensaje  
 Transport.*send*(message);  
  
 }  
  
 // Mensaje correcto  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Correo enviado.", Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 } catch (AddressException e) {  
 e.printStackTrace();  
 } catch (MessagingException e) {  
 e.printStackTrace();  
 }  
 }else{  
 // Mensaje de error  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Correo no enviado." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
 }  
  
 public BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener  
 = new BottomNavigationView.OnNavigationItemSelectedListener() {  
  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*navigation\_llamar*:  
 llamar();  
 return true;  
 case R.id.*navigation\_maps*:  
 maps();  
 return true;  
 }  
 return false;  
 }  
 };  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 // Creamos el menu  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 // Creamos un intento  
 Intent i;  
  
 // Depende de lo que pulse hace una opcion  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 // Creamos un intento  
 Intent i;  
  
 // Depende de lo que pulse hace una opcion  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos el registro  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 // Inicializamos el intento  
 i = new Intent(this, paginaPrincipal.class);  
 // Emepezamos la activity  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
  
 */\*\*  
 \* Se hace una llamada al contacto  
 \*/* @SuppressLint("MissingPermission")  
 public void hacerLlamada() {  
 // Hacemos un intento de llamada  
 Intent i=new Intent(Intent.*ACTION\_CALL*);  
 // Llamamos al telefono  
 i.setData(Uri.*parse*("tel:"+telefono));  
 // Inicializmaos la activitad  
 startActivity(i);  
 }  
  
 */\*\*  
 \* Compruebo los permisos  
 \** ***@param*** *context  
 \** ***@param*** *permissions  
 \** ***@return*** *\*/* private static boolean hasPermissions(Context context, String... permissions) {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M* && context != null && permissions != null) {  
 for (String permission : permissions) {  
 if (ActivityCompat.*checkSelfPermission*(context, permission) != PackageManager.*PERMISSION\_GRANTED*) {  
 return false;  
 }  
 }  
 }  
 return true;  
 }  
  
 */\*\*  
 \* Compruebo los permisos aceptados  
 \** ***@param*** *requestCode  
 \** ***@param*** *permissions  
 \** ***@param*** *grantResults  
 \*/* @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 final int REQUEST = 112;  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 switch (requestCode) {  
 case REQUEST: {  
 if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 hacerLlamada();  
 } else {  
 Toast.*makeText*(animal.this, "The app was not allowed to call.", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
 }  
  
 */\*\*  
 \* Comprueba que tod esta listo para llamar  
 \*/* public void llamar(){  
  
 final int REQUEST = 112;  
 if (Build.VERSION.*SDK\_INT* >= 23) {  
 // Pedimos los permisos  
 String[] PERMISSIONS = {android.Manifest.permission.*CALL\_PHONE*};  
 if (!*hasPermissions*(animal.this, PERMISSIONS)) {  
 ActivityCompat.*requestPermissions*(animal.this, PERMISSIONS, REQUEST );  
 } else {  
 hacerLlamada();  
 }  
 } else {  
 hacerLlamada();  
 }  
 }  
  
 */\*\*  
 \* Ver la posicion en el mapa.  
 \*/* public void maps(){  
 Uri gmmIntentUri = Uri.*parse*("geo:0,0?q="+direccion);  
 Intent mapIntent = new Intent(Intent.*ACTION\_VIEW*, gmmIntentUri);  
 mapIntent.setPackage("com.google.android.apps.maps");  
 startActivity(mapIntent);  
 }  
}

DI5. Ayuda.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Typeface;  
import android.os.Bundle;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.TextView;  
  
public class ayuda extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // TextView de la ayuda  
 TextView enunciadoAyuda, txtAyuda;  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_ayuda*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos los widgets  
 enunciadoAyuda = findViewById(R.id.*enunciadoAyuda*);  
 txtAyuda = findViewById(R.id.*txtAyuda*);  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 // Ponemos las fuentes  
 enunciadoAyuda.setTypeface(enunciado);  
 txtAyuda.setTypeface(texto);  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
}

DI6. Contacto.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Typeface;  
import android.os.Bundle;  
import android.os.StrictMode;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.Properties;  
  
import javax.mail.Authenticator;  
import javax.mail.Message;  
import javax.mail.MessagingException;  
import javax.mail.PasswordAuthentication;  
import javax.mail.Session;  
import javax.mail.Transport;  
import javax.mail.internet.AddressException;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;  
  
public class contacto extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // TextView de contacto  
 TextView enunciadoContacto, txtNombreContacto, txtEmail, txtMensaje;  
  
 // Button de contacto  
 Button btnEnviarContacto;  
  
 // EditText de contacto  
 EditText edtMensaje, edtEmail, edtNombreContacto;  
  
 // Declaro variables  
 String correo;  
 String password;  
  
 // Declaro los widget  
 Button enviar;  
  
 // Creo una sesion  
 Session session;  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_contacto*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 // Inicializamos los widgets  
 enunciadoContacto = findViewById(R.id.*enunciadoContacto*);  
 txtNombreContacto = findViewById(R.id.*txtNombreContacto*);  
 txtEmail = findViewById(R.id.*txtEmail*);  
 txtMensaje = findViewById(R.id.*txtMensaje*);  
 edtMensaje = findViewById(R.id.*edtMensaje*);  
 edtEmail = findViewById(R.id.*edtEmail*);  
 edtNombreContacto = findViewById(R.id.*edtNombreContacto*);  
 btnEnviarContacto = findViewById(R.id.*btnEnviarContacto*);  
  
 // Ponemos las fuentes  
 enunciadoContacto.setTypeface(enunciado);  
 txtNombreContacto.setTypeface(texto);  
 txtEmail.setTypeface(texto);  
 txtMensaje.setTypeface(texto);  
 btnEnviarContacto.setTypeface(texto);  
 }  
  
 public boolean compararNombre(){  
  
 String nombre;  
 Boolean correcto;  
  
 nombre = edtNombreContacto.getText().toString();  
 correcto = false;  
  
 if(!nombre.isEmpty() && nombre.length()>4){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el nombre." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean compararEmail(){  
  
 String email;  
 Boolean correcto;  
  
 email = edtEmail.getText().toString();  
 correcto = false;  
  
 if(!email.isEmpty() &&  
 android.util.Patterns.*EMAIL\_ADDRESS*.matcher(email).matches()){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el email." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean compararMensaje(){  
  
 String mensaje;  
 Boolean correcto;  
  
 mensaje = edtMensaje.getText().toString();  
 correcto = false;  
  
 if(!mensaje.isEmpty()){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el mensaje." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
  
 public void enviarCorreo(View v){  
  
 if(compararNombre() && compararEmail() && compararMensaje()){  
 // pongo email y contraseña  
 correo = "adoptalescontacto@gmail.com";  
 password = "Ru83nP1l0m03105";  
  
 // Ponemos las politica  
 StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();  
 StrictMode.*setThreadPolicy*(policy);  
 // Creamos unas propiedades  
 Properties properties = new Properties();  
 // ponemos la conexion  
 properties.put("mail.smtp.auth", "true");  
 properties.put("mail.smtp.starttls.enable", "true");  
 properties.put("mail.smtp.host", "smtp.gmail.com");  
 properties.put("mail.smtp.port", "587");  
  
 // intentamos  
 try {  
 // Inicializamos la sesion  
 session = Session.*getDefaultInstance*(properties, new Authenticator() {  
 @Override  
 // Autenticacion  
 protected PasswordAuthentication getPasswordAuthentication() {  
 return new PasswordAuthentication(correo, password);  
 }  
 });  
  
 // Si la sesion es erronea  
 if (session != null) {  
 // Iniciamos sesion  
 Message message = new MimeMessage(session);  
  
 // Quien lo envia  
 message.setFrom(new InternetAddress(correo));  
  
 // Titulo  
 message.setSubject("Mensaje de contacto");  
  
 // Quien lo recibe  
 message.setRecipients(Message.RecipientType.*TO*, InternetAddress.*parse*(correo));  
  
 // Mensaje y su forma  
 message.setContent("Buenas, soy " + edtNombreContacto.getText().toString() +  
 "<br>Email: " + edtEmail.getText().toString() +  
 "<br>Te quiero decir: "  
 + edtMensaje.getText().toString(), "text/html; charset=utf-8");  
  
 // Enviamos el mensaje  
 Transport.*send*(message);  
  
 }  
  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Correo enviado.", Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 } catch (AddressException e) {  
 e.printStackTrace();  
 } catch (MessagingException e) {  
 e.printStackTrace();  
 }  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Correo no enviado." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
}

DI7. Cuenta.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Bitmap;  
import android.graphics.Color;  
import android.graphics.Typeface;  
import android.graphics.drawable.BitmapDrawable;  
import android.graphics.drawable.Drawable;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.support.design.widget.FloatingActionButton;  
import android.support.design.widget.NavigationView;  
import android.support.design.widget.Snackbar;  
import android.support.v4.graphics.drawable.RoundedBitmapDrawable;  
import android.support.v4.graphics.drawable.RoundedBitmapDrawableFactory;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.util.Log;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.GridView;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.TextView;  
  
import com.squareup.picasso.Picasso;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class cuenta extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // TextView en tu cuenta  
 TextView txtNombreCuenta, txtEmailCuenta, txtIdCuenta;  
  
 // ImageView de tu cuenta  
 CircleImageView imgCuenta;  
  
 // ListView de tu cuenta  
 GridView listAnimalesCuenta;  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 // URL to get contacts JSON  
 private static String *url*, *urlAnimal*;  
  
 // JSON Node names  
 private static final String *TAG\_USERINFO* = "usuario";  
 private static final String *TAG\_ID* = "id";  
 private static final String *TAG\_NOMBRE* = "nombre";  
 private static final String *TAG\_NOMBREUSER* = "nombreuser";  
 private static final String *TAG\_IMG* = "imgperfil";  
 private static final String *TAG\_TELEFONO* = "telefono";  
 private static final String *TAG\_EMAIL* = "email";  
 private static final String *TAG\_DIRECCION* = "direccion";  
  
 // JSON Node names  
 private static final String *TAG\_ANIMALESINFO* = "animales";  
 private static final String *TAG\_IDANIMAL* = "id";  
 private static final String *TAG\_NOMBREANIMAL* = "nombre";  
 private static final String *TAG\_RAZA* = "raza";  
 private static final String *TAG\_TIPO* = "tipo";  
 private static final String *TAG\_DESCRIPCION* = "descripcion";  
 private static final String *TAG\_IMGANIMAL* = "imganimal";  
 private static final String *TAG\_EDAD* = "edad";  
 private static final String *TAG\_IDUSER* = "iduser";  
 private static final String *TAG\_GENERO* = "genero";  
 private static final String *TAG\_CHIP* = "chip";  
 private static final String *TAG\_VACUNA* = "vacuna";  
  
 String id ,nombre, nombreuser, imgperfil, telefono, email, direccion;  
  
 String [] nomAnimal;  
 String [] idAnimal;  
 String [] razaAnimal;  
 String [] tipoAnimal;  
 String [] descripcionAnimal;  
 String [] imgAnimal;  
 String [] edadAnimal;  
 String [] iduserAnimal;  
 String [] chipAnimal;  
 String [] generoAnimal;  
 String [] vacunaAnimal;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_cuenta*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.*fab*);  
 fab.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent i = new Intent(cuenta.this, subiranimal.class);  
 i.putExtra("iduser", id);  
 startActivity(i);  
 }  
 });  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 // Inicializamos los widgets  
 txtNombreCuenta = findViewById(R.id.*txtNombreCuenta*);  
 txtEmailCuenta = findViewById(R.id.*txtEmailCuenta*);  
 txtIdCuenta = findViewById(R.id.*txtIdCuenta*);  
 imgCuenta = findViewById(R.id.*imgCuenta*);  
 listAnimalesCuenta = findViewById(R.id.*listAnimalesCuenta*);  
  
 // Ponemos las fuentes  
 txtNombreCuenta.setTypeface(enunciado);  
 txtEmailCuenta.setTypeface(texto);  
  
 String email = mPrefs.getString("emailuser", "no email");  
 String password = mPrefs.getString("password", "no password");  
  
 *url* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/datosTuCuenta.php?email="+email+"&password="+ password;  
 // Ejecutamos el metodo GetMeme  
 new GetDatos().execute();  
  
 }  
  
  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/* private class GetDatos extends AsyncTask<Void, Void, Void> {  
  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> studentList;  
  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 // Creating service handler class instance  
 WebRequest webreq = new WebRequest();  
  
 // Making a request to url and getting response  
 String jsonStr = webreq.makeWebServiceCall(*url*, WebRequest.*GET*);  
  
 Log.*d*("Response: ", "> " + jsonStr);  
  
 studentList = ParseJSON(jsonStr);  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 txtIdCuenta.setText("#"+id);  
 txtNombreCuenta.setText(nombreuser);  
 txtEmailCuenta.setText(email);  
  
  
 Log.*e*("Response: ", "> " + imgperfil);  
  
 Picasso.*with*(cuenta.this)  
 .load("https://www.losnaranjosdam.online/2dam/ruben/adoptales/avatar/"+ imgperfil)  
 .error(R.mipmap.*ic\_launcher*)  
 .into(imgCuenta);  
  
 *urlAnimal* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/todosTusAnimales.php?iduser="+id;  
 // Ejecutamos el metodo GetMeme  
 new GetAnimal().execute();  
 }  
 }  
  
 private ArrayList<HashMap<String, String>> ParseJSON(String json) {  
 if (json != null) {  
 try {  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> usuarioList = new ArrayList<>();  
  
 JSONObject jsonObj = new JSONObject(json);  
  
 // Getting JSON Array node  
 JSONArray animales = jsonObj.getJSONArray(*TAG\_USERINFO*);  
  
 // looping through All Students  
 for (int i = 0; i < animales.length(); i++) {  
 JSONObject c = animales.getJSONObject(i);  
  
 nombre = c.getString(*TAG\_NOMBRE*);  
 nombreuser = c.getString(*TAG\_NOMBREUSER*);  
 id = c.getString(*TAG\_ID*);  
 telefono = c.getString(*TAG\_TELEFONO*);  
 email = c.getString(*TAG\_EMAIL*);  
 direccion = c.getString(*TAG\_DIRECCION*);  
 imgperfil = c.getString(*TAG\_IMG*);  
  
 }  
  
 return usuarioList;  
 } catch (JSONException e) {  
 e.printStackTrace();  
 return null;  
 }  
 } else {  
 Log.*e*("ServiceHandler", "Couldn't get any data from the url");  
 return null;  
 }  
 }  
  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/* private class GetAnimal extends AsyncTask<Void, Void, Void> {  
  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> studentList;  
  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 // Creating service handler class instance  
 WebRequest webreq = new WebRequest();  
  
 // Making a request to url and getting response  
 String jsonStr = webreq.makeWebServiceCall(*urlAnimal*, WebRequest.*GET*);  
  
 Log.*d*("Response: ", "> " + jsonStr);  
  
 studentList = ParseJSON2(jsonStr);  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 try {  
 AdaptadorTusAnimales adapter = new AdaptadorTusAnimales(cuenta.this, nomAnimal, imgAnimal, texto);  
 listAnimalesCuenta.setAdapter(adapter);  
 listAnimalesCuenta.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View v, int posicion, long id) {  
 // Creo un intento  
 Intent in = new Intent(listAnimalesCuenta.getContext(), tuanimal.class);  
  
 in.putExtra("id", idAnimal[posicion]);  
 in.putExtra("nombre", nomAnimal[posicion]);  
 in.putExtra("raza", razaAnimal[posicion]);  
 in.putExtra("tipo", tipoAnimal[posicion]);  
 in.putExtra("descripcion", descripcionAnimal[posicion]);  
 in.putExtra("imganimal", imgAnimal[posicion]);  
 in.putExtra("edad", edadAnimal[posicion]);  
 in.putExtra("iduser", iduserAnimal[posicion]);  
 in.putExtra("genero", generoAnimal[posicion]);  
 if(Integer.*parseInt*(chipAnimal[posicion]) == 1){  
 in.putExtra("chip", "Si");  
 }else{  
 in.putExtra("chip", "No");  
 }  
 if(Integer.*parseInt*(vacunaAnimal[posicion]) == 1){  
 in.putExtra("vacuna", "Si");  
 }else{  
 in.putExtra("vacuna", "No");  
 }  
  
 in.putExtra("telefono", telefono);  
 in.putExtra("email", email);  
  
 // empiezo activity  
 startActivity(in);  
 }  
 });  
 } catch (Exception e) {  
 listAnimalesCuenta.setAdapter(null);  
 }  
 }  
 }  
  
 private ArrayList<HashMap<String, String>> ParseJSON2(String json) {  
 if (json != null) {  
 try {  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> usuarioList = new ArrayList<>();  
  
 JSONObject jsonObj = new JSONObject(json);  
  
 // Getting JSON Array node  
 JSONArray animales = jsonObj.getJSONArray(*TAG\_ANIMALESINFO*);  
  
 idAnimal = new String[animales.length()];  
 nomAnimal = new String[animales.length()];  
 razaAnimal = new String[animales.length()];  
 tipoAnimal = new String[animales.length()];  
 descripcionAnimal = new String[animales.length()];  
 edadAnimal = new String[animales.length()];  
 iduserAnimal = new String[animales.length()];  
 imgAnimal = new String[animales.length()];  
 vacunaAnimal = new String[animales.length()];  
 generoAnimal = new String[animales.length()];  
 chipAnimal = new String[animales.length()];  
  
 // looping through All Students  
 for (int i = 0; i < animales.length(); i++) {  
 JSONObject c = animales.getJSONObject(i);  
  
 String id = c.getString(*TAG\_IDANIMAL*);  
 String nombre = c.getString(*TAG\_NOMBREANIMAL*);  
 String raza = c.getString(*TAG\_RAZA*);  
 String tipo = c.getString(*TAG\_TIPO*);  
 String descripcion = c.getString(*TAG\_DESCRIPCION*);  
 String imganimal = c.getString(*TAG\_IMGANIMAL*);  
 String edad = c.getString(*TAG\_EDAD*);  
 String iduser = c.getString(*TAG\_IDUSER*);  
 String vacuna = c.getString(*TAG\_VACUNA*);  
 String chip = c.getString(*TAG\_CHIP*);  
 String genero = c.getString(*TAG\_GENERO*);  
  
 // tmp hashmap for single student  
 HashMap<String, String> animal = new HashMap<String, String>();  
  
 // adding each child node to HashMap key => value  
 animal.put(*TAG\_ID*, id);  
 animal.put(*TAG\_NOMBRE*, nombre);  
 animal.put(*TAG\_RAZA*, imganimal);  
 animal.put(*TAG\_TIPO*, id);  
 animal.put(*TAG\_DESCRIPCION*, nombre);  
 animal.put(*TAG\_IMGANIMAL*, imganimal);  
 animal.put(*TAG\_EDAD*, nombre);  
 animal.put(*TAG\_IDUSER*, imganimal);  
 animal.put(*TAG\_VACUNA*, vacuna);  
 animal.put(*TAG\_CHIP*, chip);  
 animal.put(*TAG\_GENERO*, genero);  
  
 idAnimal[i] = id;  
 nomAnimal[i] = nombre;  
 razaAnimal[i] = raza;  
 tipoAnimal[i] = tipo;  
 descripcionAnimal[i] = descripcion;  
 edadAnimal[i] = edad;  
 imgAnimal[i] = imganimal;  
 iduserAnimal[i] = iduser;  
 vacunaAnimal[i] = vacuna;  
 chipAnimal[i] = chip;  
 generoAnimal[i] = genero;  
 }  
  
 return usuarioList;  
 } catch (JSONException e) {  
 e.printStackTrace();  
 return null;  
 }  
 } else {  
 Log.*e*("ServiceHandler", "Couldn't get any data from the url");  
 return null;  
 }  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
}

DI8. Info.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Typeface;  
import android.os.Bundle;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.TextView;  
  
public class info extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // TextView en informacion  
 TextView enunciadoInfo, txtAyuda;  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_info*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 // Inicializamos los widgets  
 enunciadoInfo = findViewById(R.id.*enunciadoInfo*);  
 txtAyuda = findViewById(R.id.*txtAyuda*);  
  
 // Ponemos las fuentes  
 enunciadoInfo.setTypeface(enunciado);  
 txtAyuda.setTypeface(texto);  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
}

DI9. Login.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Typeface;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.util.Log;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
public class login extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // TextView de login  
 TextView enunciadoLogin, txtEmailLogin, txtPasswordLogin;  
  
 // EditText de login  
 EditText edtEmailLogin, edtPasswordLogin;  
  
 // Button de login  
 Button btnEntrarLogin;  
  
 // URL to get contacts JSON  
 private static String *url* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/todosAnimales.php";  
  
 // JSON Node names  
 private static final String *TAG\_ANIMALESINFO* = "animales";  
 private static final String *TAG\_ID* = "id";  
 private static final String *TAG\_NOMBRE* = "nombre";  
 private static final String *TAG\_RAZA* = "raza";  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 // Inicializamos todos los widgets  
 enunciadoLogin = findViewById(R.id.*enunciadoLogin*);  
 txtEmailLogin = findViewById(R.id.*txtEmailLogin*);  
 txtPasswordLogin = findViewById(R.id.*txtPasswordLogin*);  
 btnEntrarLogin = findViewById(R.id.*btnEntrarLogin*);  
 edtEmailLogin = findViewById(R.id.*edtEmailLogin*);  
 edtPasswordLogin = findViewById(R.id.*edtPasswordLogin*);  
  
 // Ponemos las fuentes  
 enunciadoLogin.setTypeface(enunciado);  
 txtEmailLogin.setTypeface(texto);  
 txtPasswordLogin.setTypeface(texto);  
 btnEntrarLogin.setTypeface(texto);  
 }  
  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/* private class GetPartida extends AsyncTask<Void, Void, Void> {  
  
 // Hashmap for ListView  
 String respuesta;  
  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 // Creating service handler class instance  
 WebRequest webreq = new WebRequest();  
  
 // Making a request to url and getting response  
 respuesta = webreq.makeWebServiceCall(*url*, WebRequest.*GET*);  
  
 Log.*d*("Response: ", ">" + respuesta);  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 if(Integer.*parseInt*(respuesta) == 1){  
 // Creamos un registro con esa id  
 SharedPreferences mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 SharedPreferences.Editor myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", true);  
 myEditor.putString("emailuser", edtEmailLogin.getText().toString());  
 myEditor.putString("password", edtPasswordLogin.getText().toString());  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 Intent i = new Intent(login.this, paginaPrincipal.class);  
 startActivity(i);  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Email o contraseña incorrecta." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
 }  
  
 }  
  
 public boolean comprobarEmail(){  
 String email;  
 Boolean correcto;  
  
 email = edtEmailLogin.getText().toString();  
 correcto = false;  
  
 if(!email.isEmpty() &&  
 android.util.Patterns.*EMAIL\_ADDRESS*.matcher(email).matches()){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el email." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean comprobarPassword(){  
 String password;  
 Boolean correcto;  
  
 password = edtPasswordLogin.getText().toString();  
 correcto = false;  
  
 if(!password.isEmpty() && password.length()>6){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en la contraseña." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public void loguearse(View v){  
  
 String email;  
 String password;  
  
 email = edtEmailLogin.getText().toString();  
 password = edtPasswordLogin.getText().toString();  
  
 if(comprobarEmail() && comprobarPassword()){  
 *url* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/login.php?email="+email+"&password="+password+"";  
  
 // Ejecutamos el metodo GetMeme  
 new GetPartida().execute();  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error al loguearse." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
}

DI10. PaginaPrincipal.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.Manifest;  
import android.app.Dialog;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Color;  
import android.graphics.Typeface;  
import android.graphics.drawable.ColorDrawable;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.support.design.widget.NavigationView;  
import android.support.v4.app.ActivityCompat;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.util.DisplayMetrics;  
import android.util.Log;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.GridView;  
import android.widget.ImageButton;  
import android.widget.ListView;  
import android.widget.RadioButton;  
import android.widget.TextView;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
public class paginaPrincipal extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // ListView pagina principal  
 GridView listAnimalesPrincipal;  
  
 String [] nomAnimal;  
 String [] idAnimal;  
 String [] razaAnimal;  
 String [] tipoAnimal;  
 String [] descripcionAnimal;  
 String [] imgAnimal;  
 String [] edadAnimal;  
 String [] iduserAnimal;  
 String [] chipAnimal;  
 String [] generoAnimal;  
 String [] vacunaAnimal;  
  
 int versionBDRefugio;  
  
 // URL to get contacts JSON  
 private static String *url* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/todosAnimales.php";  
  
 // JSON Node names  
 private static final String *TAG\_ANIMALESINFO* = "animales";  
 private static final String *TAG\_ID* = "id";  
 private static final String *TAG\_NOMBRE* = "nombre";  
 private static final String *TAG\_RAZA* = "raza";  
 private static final String *TAG\_TIPO* = "tipo";  
 private static final String *TAG\_DESCRIPCION* = "descripcion";  
 private static final String *TAG\_IMGANIMAL* = "imganimal";  
 private static final String *TAG\_EDAD* = "edad";  
 private static final String *TAG\_IDUSER* = "iduser";  
 private static final String *TAG\_GENERO* = "genero";  
 private static final String *TAG\_CHIP* = "chip";  
 private static final String *TAG\_VACUNA* = "vacuna";  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 ImageButton btnBuscar;  
  
 Dialog filtros;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_pagina\_principal*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 filtros = new Dialog(this);  
  
 // Pedimos los permisos  
 final int REQUEST = 112;  
 String[] PERMISSIONS = {android.Manifest.permission.*CALL\_PHONE*, Manifest.permission.*INTERNET*};  
 ActivityCompat.*requestPermissions*(this, PERMISSIONS, REQUEST );  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos los widgets  
 listAnimalesPrincipal = findViewById(R.id.*listAnimalesPrincipal*);  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 btnBuscar = findViewById(R.id.*btnBuscar*);  
  
 // Ejecutamos el metodo GetMeme  
 new GetPartida().execute();  
 }  
  
 public void VerFiltros(View v){  
 filtros.setContentView(R.layout.*contenido\_filtros*);  
  
 Button btnAtrasFiltros = filtros.findViewById(R.id.*btnAtrasFiltros*);  
 Button btnBuscarAnimal = filtros.findViewById(R.id.*btnBuscarAnimal*);  
  
 TextView enunciadoBuscar = filtros.findViewById(R.id.*enunciadoBuscar*);  
 TextView txtNombreAnimalBuscar = filtros.findViewById(R.id.*txtNombreAnimalBuscar*);  
 TextView txtTipoAnimalBuscar = filtros.findViewById(R.id.*txtTipoAnimalBuscar*);  
 TextView txtRazaAnimalBuscar = filtros.findViewById(R.id.*txtRazaAnimalBuscar*);  
 TextView txtEdadAnimalBuscar = filtros.findViewById(R.id.*txtEdadAnimalBuscar*);  
 TextView txtGeneroAnimalBuscar = filtros.findViewById(R.id.*txtGeneroAnimalBuscar*);  
 TextView txtVacunaAnimalBuscar = filtros.findViewById(R.id.*txtVacunaAnimalBuscar*);  
 TextView txtChipAnimalBuscar = filtros.findViewById(R.id.*txtChipAnimalBuscar*);  
  
 // EditText de contacto  
 EditText edtNombreAnimalBuscar = filtros.findViewById(R.id.*edtNombreAnimalBuscar*);  
 EditText edtRazaAnimalBuscar = filtros.findViewById(R.id.*edtRazaAnimalBuscar*);  
 EditText edtEdadAnimalBuscar = filtros.findViewById(R.id.*edtEdadAnimalBuscar*);  
  
 RadioButton rbGeneroMAnimalBuscar = filtros.findViewById(R.id.*rbGeneroMAnimalBuscar*);  
 RadioButton rbGeneroHAnimalBuscar = filtros.findViewById(R.id.*rbGeneroHAnimalBuscar*);  
 RadioButton rbVacunaSAnimalBuscar = filtros.findViewById(R.id.*rbVacunaSAnimalBuscar*);  
 RadioButton rbVacunaNAnimalBuscar = filtros.findViewById(R.id.*rbVacunaNAnimalBuscar*);  
 RadioButton rbChipSAnimalBuscar = filtros.findViewById(R.id.*rbChipSAnimalBuscar*);  
 RadioButton rbChipNAnimalBuscar = filtros.findViewById(R.id.*rbChipNAnimalBuscar*);  
  
 enunciadoBuscar.setTypeface(enunciado);  
 txtNombreAnimalBuscar.setTypeface(texto);  
 txtTipoAnimalBuscar.setTypeface(texto);  
 txtRazaAnimalBuscar.setTypeface(texto);  
 txtEdadAnimalBuscar.setTypeface(texto);  
 txtGeneroAnimalBuscar.setTypeface(texto);  
 txtVacunaAnimalBuscar.setTypeface(texto);  
 txtChipAnimalBuscar.setTypeface(texto);  
 rbGeneroMAnimalBuscar.setTypeface(texto);  
 rbGeneroHAnimalBuscar.setTypeface(texto);  
 rbVacunaSAnimalBuscar.setTypeface(texto);  
 rbVacunaNAnimalBuscar.setTypeface(texto);  
 rbChipSAnimalBuscar.setTypeface(texto);  
 rbChipNAnimalBuscar.setTypeface(texto);  
 btnBuscarAnimal.setTypeface(texto);  
 btnAtrasFiltros.setTypeface(texto);  
  
 btnAtrasFiltros.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v) {  
 filtros.dismiss();  
 }  
 });  
  
 filtros.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*));  
 filtros.show();  
 }  
  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/* private class GetPartida extends AsyncTask<Void, Void, Void> {  
  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> studentList;  
  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 // Creating service handler class instance  
 WebRequest webreq = new WebRequest();  
  
 // Making a request to url and getting response  
 String jsonStr = webreq.makeWebServiceCall(*url*, WebRequest.*GET*);  
  
 Log.*d*("Response: ", "> " + jsonStr);  
  
 studentList = ParseJSON(jsonStr);  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 try {  
  
 DisplayMetrics metrics = new DisplayMetrics();  
 getWindowManager().getDefaultDisplay().getMetrics(metrics);  
 int width = metrics.widthPixels;  
 int height = metrics.heightPixels;  
  
 AdaptadorAnimales adapter = new AdaptadorAnimales(paginaPrincipal.this, nomAnimal, imgAnimal, texto);  
 listAnimalesPrincipal.setAdapter(adapter);  
 listAnimalesPrincipal.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View v, int posicion, long id) {  
 // Creo un intento  
 Intent in = new Intent(listAnimalesPrincipal.getContext(), animal.class);  
  
 in.putExtra("nombre", nomAnimal[posicion]);  
 in.putExtra("raza", razaAnimal[posicion]);  
 in.putExtra("tipo", tipoAnimal[posicion]);  
 in.putExtra("descripcion", descripcionAnimal[posicion]);  
 in.putExtra("edad", edadAnimal[posicion]);  
 in.putExtra("iduser", iduserAnimal[posicion]);  
 in.putExtra("imgAnimal", imgAnimal[posicion]);  
 in.putExtra("genero", generoAnimal[posicion]);  
 if(Integer.*parseInt*(chipAnimal[posicion]) == 1){  
 in.putExtra("chip", "Si");  
 }else{  
 in.putExtra("chip", "No");  
 }  
 if(Integer.*parseInt*(vacunaAnimal[posicion]) == 1){  
 in.putExtra("vacuna", "Si");  
 }else{  
 in.putExtra("vacuna", "No");  
 }  
  
 // empiezo activity  
 startActivity(in);  
 }  
 });  
 } catch (Exception e) {  
 listAnimalesPrincipal.setAdapter(null);  
 }  
 }  
  
 }  
  
 private ArrayList<HashMap<String, String>> ParseJSON(String json) {  
 if (json != null) {  
 try {  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> animalList = new ArrayList<HashMap<String, String>>();  
  
  
  
 JSONObject jsonObj = new JSONObject(json);  
  
 // Getting JSON Array node  
 JSONArray animales = jsonObj.getJSONArray(*TAG\_ANIMALESINFO*);  
  
 idAnimal = new String[animales.length()];  
 nomAnimal = new String[animales.length()];  
 razaAnimal = new String[animales.length()];  
 tipoAnimal = new String[animales.length()];  
 descripcionAnimal = new String[animales.length()];  
 edadAnimal = new String[animales.length()];  
 iduserAnimal = new String[animales.length()];  
 imgAnimal = new String[animales.length()];  
 vacunaAnimal = new String[animales.length()];  
 generoAnimal = new String[animales.length()];  
 chipAnimal = new String[animales.length()];  
  
 // looping through All Students  
 for (int i = 0; i < animales.length(); i++) {  
 JSONObject c = animales.getJSONObject(i);  
  
 String id = c.getString(*TAG\_ID*);  
 String nombre = c.getString(*TAG\_NOMBRE*);  
 String raza = c.getString(*TAG\_RAZA*);  
 String tipo = c.getString(*TAG\_TIPO*);  
 String descripcion = c.getString(*TAG\_DESCRIPCION*);  
 String imganimal = c.getString(*TAG\_IMGANIMAL*);  
 String edad = c.getString(*TAG\_EDAD*);  
 String iduser = c.getString(*TAG\_IDUSER*);  
 String vacuna = c.getString(*TAG\_VACUNA*);  
 String chip = c.getString(*TAG\_CHIP*);  
 String genero = c.getString(*TAG\_GENERO*);  
  
 // tmp hashmap for single student  
 HashMap<String, String> animal = new HashMap<String, String>();  
  
 // adding each child node to HashMap key => value  
 animal.put(*TAG\_ID*, id);  
 animal.put(*TAG\_NOMBRE*, nombre);  
 animal.put(*TAG\_RAZA*, imganimal);  
 animal.put(*TAG\_TIPO*, id);  
 animal.put(*TAG\_DESCRIPCION*, nombre);  
 animal.put(*TAG\_IMGANIMAL*, imganimal);  
 animal.put(*TAG\_EDAD*, nombre);  
 animal.put(*TAG\_IDUSER*, imganimal);  
 animal.put(*TAG\_VACUNA*, vacuna);  
 animal.put(*TAG\_CHIP*, chip);  
 animal.put(*TAG\_GENERO*, genero);  
  
 idAnimal[i] = id;  
 nomAnimal[i] = nombre;  
 razaAnimal[i] = raza;  
 tipoAnimal[i] = tipo;  
 descripcionAnimal[i] = descripcion;  
 edadAnimal[i] = edad;  
 iduserAnimal[i] = iduser;  
 imgAnimal[i] = imganimal;  
 vacunaAnimal[i] = vacuna;  
 chipAnimal[i] = chip;  
 generoAnimal[i] = genero;  
  
 // adding student to students list  
 animalList.add(animal);  
 }  
  
 return animalList;  
 } catch (JSONException e) {  
 e.printStackTrace();  
 return null;  
 }  
 } else {  
 Log.*e*("ServiceHandler", "Couldn't get any data from the url");  
 return null;  
 }  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
  
 return true;  
 }  
}

DI11. Refugio.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.annotation.SuppressLint;  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.content.pm.PackageManager;  
import android.graphics.Typeface;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Bundle;  
import android.support.annotation.NonNull;  
import android.support.design.widget.BottomNavigationView;  
import android.support.design.widget.NavigationView;  
import android.support.v4.app.ActivityCompat;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.squareup.picasso.Picasso;  
  
public class refugio extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // TextView del refugio  
 TextView txtNombreRefugio, txtNumeroRefugio, txtDireccionRefugio, txtEmailRefugio, txtInformacion  
 , txtEmail ,txtMensaje;  
  
 // ImageView del refugio  
 ImageView imgRefugio;  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 String idRefugio;  
 String nomRefugio;  
 String imgRef;  
 String emailRefugio;  
 String telefonoRefugio;  
 String direccionRefugio;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_refugio*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.*navigation*);  
 navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 // Inicializamos los widgets  
 txtNombreRefugio = findViewById(R.id.*txtNombreRefugio*);  
 txtNumeroRefugio = findViewById(R.id.*txtNumeroRefugio*);  
 txtDireccionRefugio = findViewById(R.id.*txtDireccionRefugio*);  
 txtEmailRefugio = findViewById(R.id.*txtEmailRefugio*);  
 imgRefugio = findViewById(R.id.*imgRefugio*);  
 txtInformacion = findViewById(R.id.*txtInformacion*);  
 txtEmail = findViewById(R.id.*txtEmail*);  
 txtMensaje = findViewById(R.id.*txtMensaje*);  
  
 // Ponemos el tipo de letra  
 txtNombreRefugio.setTypeface(enunciado);  
 txtEmailRefugio.setTypeface(texto);  
 txtInformacion.setTypeface(texto);  
 txtEmail.setTypeface(texto);  
 txtMensaje.setTypeface(texto);  
  
 // Cogemos los extras que nos pasan en la anterio activity  
 Bundle extras = getIntent().getExtras();  
  
 // Si los extras no son nulos  
 if (extras != null) {  
 // Guardo los datos que tengan como clave ano  
 idRefugio = extras.getString("id");  
 nomRefugio = extras.getString("nombre");  
 imgRef = extras.getString("img");  
 emailRefugio = extras.getString("email");  
 telefonoRefugio = extras.getString("telefono");  
 direccionRefugio = extras.getString("direccion");  
 } else {  
 // Se queda vacio  
 idRefugio = "";  
 nomRefugio = "";  
 imgRef = "";  
 emailRefugio = "";  
 telefonoRefugio = "";  
 direccionRefugio = "";  
 }  
  
 // Ponemos las fuentes  
 txtNombreRefugio.setText(nomRefugio);  
 txtNumeroRefugio.setText("-"+ telefonoRefugio);  
 txtDireccionRefugio.setText("-"+direccionRefugio);  
 txtEmailRefugio.setText(emailRefugio);  
  
 // Cogemos y ponemos la imagen del servidor  
 Picasso.*with*(this)  
 .load("https://www.losnaranjosdam.online/2dam/ruben/adoptales/fotosRefugio/"+ imgRef)  
 .error(R.mipmap.*ic\_launcher*)  
 .into(imgRefugio);  
 }  
  
 public BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener  
 = new BottomNavigationView.OnNavigationItemSelectedListener() {  
  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*navigation\_llamar*:  
 llamar();  
 return true;  
 case R.id.*navigation\_maps*:  
 maps();  
 return true;  
 }  
 return false;  
 }  
 };  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugio.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
  
 */\*\*  
 \* Se hace una llamada al contacto  
 \*/* @SuppressLint("MissingPermission")  
 public void hacerLlamada() {  
 // Hacemos un intento de llamada  
 Intent i=new Intent(Intent.*ACTION\_CALL*);  
 // Llamamos al telefono  
 i.setData(Uri.*parse*("tel:"+telefonoRefugio));  
 // Inicializmaos la activitad  
 startActivity(i);  
 }  
  
 */\*\*  
 \* Compruebo los permisos  
 \** ***@param*** *context  
 \** ***@param*** *permissions  
 \** ***@return*** *\*/* private static boolean hasPermissions(Context context, String... permissions) {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M* && context != null && permissions != null) {  
 for (String permission : permissions) {  
 if (ActivityCompat.*checkSelfPermission*(context, permission) != PackageManager.*PERMISSION\_GRANTED*) {  
 return false;  
 }  
 }  
 }  
 return true;  
 }  
  
 */\*\*  
 \* Compruebo los permisos aceptados  
 \** ***@param*** *requestCode  
 \** ***@param*** *permissions  
 \** ***@param*** *grantResults  
 \*/* @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 final int REQUEST = 112;  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 switch (requestCode) {  
 case REQUEST: {  
 if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 hacerLlamada();  
 } else {  
 Toast.*makeText*(refugio.this, "The app was not allowed to call.", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
 }  
  
 */\*\*  
 \* Comprueba que tod esta listo para llamar  
 \*/* public void llamar(){  
  
 final int REQUEST = 112;  
 if (Build.VERSION.*SDK\_INT* >= 23) {  
 // Pedimos los permisos  
 String[] PERMISSIONS = {android.Manifest.permission.*CALL\_PHONE*};  
 if (!*hasPermissions*(refugio.this, PERMISSIONS)) {  
 ActivityCompat.*requestPermissions*(refugio.this, PERMISSIONS, REQUEST );  
 } else {  
 hacerLlamada();  
 }  
 } else {  
 hacerLlamada();  
 }  
 }  
  
 */\*\*  
 \* Ver la posicion en el mapa.  
 \*/* public void maps(){  
 Uri gmmIntentUri = Uri.*parse*("geo:0,0?q="+direccionRefugio);  
 Intent mapIntent = new Intent(Intent.*ACTION\_VIEW*, gmmIntentUri);  
 mapIntent.setPackage("com.google.android.apps.maps");  
 startActivity(mapIntent);  
 }  
}

DI12. Refugios.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Typeface;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.util.Log;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.GridView;  
import android.widget.ListView;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
public class refugios extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // ListView de refugios  
 GridView listRefugios;  
  
 // Datos refugios  
 String [] idRefugio;  
 String [] nomRefugio;  
 String [] imgRefugio;  
 String [] emailRefugio;  
 String [] telefonoRefugio;  
 String [] direccionRefugio;  
  
 int versionBDRefugio;  
  
 // URL to get contacts JSON  
 private static String *url* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/todosRefugios.php";  
  
 // JSON Node names  
 private static final String *TAG\_STUDENTINFO* = "refugios";  
 private static final String *TAG\_ID* = "id";  
 private static final String *TAG\_NAME* = "nombrerefugio";  
 private static final String *TAG\_IMG* = "imgperfil";  
 private static final String *TAG\_EMAIL* = "email";  
 private static final String *TAG\_TELEFONO* = "telefono";  
 private static final String *TAG\_DIRECCION* = "direccion";  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_refugios*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos los widgets  
 listRefugios = findViewById(R.id.*listRefugios*);  
 // Ejecutamos el metodo GetMeme  
  
 // Creamos un registro con esa id  
 SharedPreferences mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 SharedPreferences.Editor myEditor = mPrefs.edit();  
  
 // Actualizmos la variable el entero con clave version  
 versionBDRefugio = mPrefs.getInt("versionbdrefugio", 0);  
  
 // Si es igual a 0  
 if(versionBDRefugio == 0) {  
 // La version se pondra a 0  
 myEditor.putInt("versionbdrefugio", 0);  
 }  
  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 new GetRegistro().execute();  
 }  
  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/* private class GetRegistro extends AsyncTask<Void, Void, Void> {  
  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> studentList;  
  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 // Creating service handler class instance  
 WebRequest webreq = new WebRequest();  
  
 // Making a request to url and getting response  
 String jsonStr = webreq.makeWebServiceCall(*url*, WebRequest.*GET*);  
  
 Log.*d*("Response: ", "> " + jsonStr);  
  
 studentList = ParseJSON(jsonStr);  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 try {  
 AdaptadorRefugio adapter = new AdaptadorRefugio(refugios.this, nomRefugio, imgRefugio, texto);  
 listRefugios.setAdapter(adapter);  
 listRefugios.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View v, int posicion, long id) {  
 // Creo un intento  
 Intent in = new Intent(listRefugios.getContext(), refugio.class);  
  
 in.putExtra("id", idRefugio[posicion]);  
 in.putExtra("nombre", nomRefugio[posicion]);  
 in.putExtra("img", imgRefugio[posicion]);  
 in.putExtra("email", emailRefugio[posicion]);  
 in.putExtra("telefono", telefonoRefugio[posicion]);  
 in.putExtra("direccion", direccionRefugio[posicion]);  
  
 // empiezo activity  
 startActivity(in);  
 }  
 });  
 } catch (Exception e) {  
 listRefugios.setAdapter(null);  
 }  
 }  
 }  
  
 private ArrayList<HashMap<String, String>> ParseJSON(String json) {  
 if (json != null) {  
 try {  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> studentList = new ArrayList<HashMap<String, String>>();  
  
 JSONObject jsonObj = new JSONObject(json);  
  
 // Getting JSON Array node  
 JSONArray students = jsonObj.getJSONArray(*TAG\_STUDENTINFO*);  
  
 nomRefugio = new String[students.length()];  
 idRefugio = new String[students.length()];  
 imgRefugio = new String[students.length()];  
 emailRefugio = new String[students.length()];  
 telefonoRefugio = new String[students.length()];  
 direccionRefugio = new String[students.length()];  
  
 // looping through All Students  
 for (int i = 0; i < students.length(); i++) {  
 JSONObject c = students.getJSONObject(i);  
  
 String id = c.getString(*TAG\_ID*);  
 String name = c.getString(*TAG\_NAME*);  
 String img = c.getString(*TAG\_IMG*);  
  
 // tmp hashmap for single student  
 HashMap<String, String> student = new HashMap<String, String>();  
  
 // adding each child node to HashMap key => value  
 student.put(*TAG\_ID*, id);  
 student.put(*TAG\_NAME*, name);  
 student.put(*TAG\_IMG*, img);  
  
 idRefugio[i] = c.getString(*TAG\_ID*);  
 nomRefugio[i] = c.getString(*TAG\_NAME*);  
 imgRefugio[i] = c.getString(*TAG\_IMG*);  
 emailRefugio[i] = c.getString(*TAG\_EMAIL*);  
 telefonoRefugio[i] = c.getString(*TAG\_TELEFONO*);  
 direccionRefugio[i] = c.getString(*TAG\_DIRECCION*);  
  
 // adding student to students list  
 studentList.add(student);  
 }  
 return studentList;  
 } catch (JSONException e) {  
 e.printStackTrace();  
 return null;  
 }  
 } else {  
 Log.*e*("ServiceHandler", "Couldn't get any data from the url");  
 return null;  
 }  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
}

DI13. Registro.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.app.Dialog;  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Bitmap;  
import android.graphics.Color;  
import android.graphics.Typeface;  
import android.graphics.drawable.ColorDrawable;  
import android.net.Uri;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.os.StrictMode;  
import android.provider.MediaStore;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.util.Base64;  
import android.util.Log;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.CheckBox;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Spinner;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.android.volley.AuthFailureError;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.StringRequest;  
import com.android.volley.toolbox.Volley;  
import com.squareup.picasso.Picasso;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.io.ByteArrayOutputStream;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Hashtable;  
import java.util.Map;  
import java.util.Properties;  
  
import javax.mail.Authenticator;  
import javax.mail.Message;  
import javax.mail.MessagingException;  
import javax.mail.PasswordAuthentication;  
import javax.mail.Session;  
import javax.mail.Transport;  
import javax.mail.internet.AddressException;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;  
  
public class registro extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener, View.OnClickListener {  
  
 // TextView enunciado  
 TextView enunciadoRegistro;  
  
 // TextView del registro  
 TextView txtNombreRegistro, txtNombreUserRegistro, txtEmailRegistro, txtPasswordRegistro  
 , txtTelefonoRegistro, txtDireccionRegistro, txtNacimientoRegistro;  
  
 // EditText del registro  
 EditText edtNombreRegistro, edtNombreUserRegistro, edtEmailRegistro, edtPasswordRegistro,edtPasswordRepiteRegistro, edtTelefonoRegistro, edtDireccionRegistro;  
  
 // Spinner del registro  
 Spinner spDiaRegistro, spMesRegistro, spAnoRegistro;  
  
 // Checbox del registro  
 CheckBox chkAceptarTerminosRegistro;  
  
 // Button del registro  
 Button btnRegistrar, btnAvatar, btnVerTerminos;  
  
 // Avatar  
 ImageView imgSubirAnimal;  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 // Declaro variables  
 String correo;  
 String password;  
  
 String contar;  
  
 // Creo una sesion  
 Session session;  
  
 // URL to get contacts JSON  
 private static String *url*;  
  
 // Subir imagen  
 private Bitmap bitmap;  
 private int PICK\_IMAGE\_REQUEST = 1;  
 private String UPLOAD\_URL ="https://www.losnaranjosdam.online/2dam/ruben/adoptales/subirFotoAvatar.php";  
 private String KEY\_IMAGEN = "foto";  
  
 Dialog terminos;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_registro*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 terminos = new Dialog(this);  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 // Inicializamos los wigdets  
 enunciadoRegistro = findViewById(R.id.*enunciadoRegistro*);  
 txtNombreRegistro = findViewById(R.id.*txtNombreRegistro*);  
 txtNombreUserRegistro = findViewById(R.id.*txtNombreUserRegistro*);  
 txtEmailRegistro = findViewById(R.id.*txtEmailRegistro*);  
 txtPasswordRegistro = findViewById(R.id.*txtPasswordRegistro*);  
 txtTelefonoRegistro = findViewById(R.id.*txtTelefonoRegistro*);  
 txtDireccionRegistro = findViewById(R.id.*txtDireccionRegistro*);  
 txtNacimientoRegistro = findViewById(R.id.*txtNacimientoRegistro*);  
 btnRegistrar = findViewById(R.id.*btnRegistrar*);  
 btnAvatar = findViewById(R.id.*btnAvatar*);  
 btnVerTerminos = findViewById(R.id.*btnVerTerminos*);  
 imgSubirAnimal = findViewById(R.id.*imgSubirAnimal*);  
 spDiaRegistro = findViewById(R.id.*spDiaRegistro*);  
 spMesRegistro = findViewById(R.id.*spMesRegistro*);  
 spAnoRegistro = findViewById(R.id.*spAnoRegistro*);  
 edtNombreRegistro = findViewById(R.id.*edtNombreRegistro*);  
 edtNombreUserRegistro = findViewById(R.id.*edtNombreUserRegistro*);  
 edtEmailRegistro = findViewById(R.id.*edtEmailRegistro*);  
 edtPasswordRegistro = findViewById(R.id.*edtPasswordRegistro*);  
 edtPasswordRepiteRegistro = findViewById(R.id.*edtPasswordRepiteRegistro*);  
 edtTelefonoRegistro = findViewById(R.id.*edtTelefonoRegistro*);  
 edtDireccionRegistro = findViewById(R.id.*edtDireccionRegistro*);  
 chkAceptarTerminosRegistro = findViewById(R.id.*chkAceptarTerminosRegistro*);  
  
 // Ponemos las fuentes  
 enunciadoRegistro.setTypeface(enunciado);  
 txtNombreRegistro.setTypeface(texto);  
 txtNombreUserRegistro.setTypeface(texto);  
 txtEmailRegistro.setTypeface(texto);  
 txtPasswordRegistro.setTypeface(texto);  
 btnVerTerminos.setTypeface(texto);  
 txtTelefonoRegistro.setTypeface(texto);  
 txtDireccionRegistro.setTypeface(texto);  
 txtNacimientoRegistro.setTypeface(texto);  
 btnRegistrar.setTypeface(texto);  
 btnAvatar.setTypeface(texto);  
 chkAceptarTerminosRegistro.setTypeface(texto);  
  
  
 // Creamos una array de años  
 String [] anos = new String[60];  
 int ano = 2003;  
 for(int i = 0; i < anos.length; i++){  
 anos[i] = String.*valueOf*(ano);  
 ano--;  
 }  
  
 // Adaptamos los datos a un spinner  
 ArrayAdapter<CharSequence> adapter = ArrayAdapter.*createFromResource*(this, R.array.*dias*, android.R.layout.*simple\_spinner\_item*);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 spDiaRegistro.setAdapter(adapter);  
  
 // Adaptamos los datos a un spinner  
 adapter = ArrayAdapter.*createFromResource*(this, R.array.*meses*, android.R.layout.*simple\_spinner\_item*);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 spMesRegistro.setAdapter(adapter);  
  
 // Adaptamos los datos a un spinner  
 spAnoRegistro.setAdapter(new ArrayAdapter<String>(this, android.R.layout.*simple\_spinner\_item*, anos));  
 btnAvatar.setOnClickListener(this);  
  
 }  
  
 public void VerTerminos(View v){  
 terminos.setContentView(R.layout.*contenido\_terminos*);  
 Button btnAtrasTerminos = terminos.findViewById(R.id.*btnAtrasTerminos*);  
 TextView txtEnunciadoTerminos = terminos.findViewById(R.id.*txtEnunciadoTerminos*);  
  
 btnAtrasTerminos.setTypeface(texto);  
 txtEnunciadoTerminos.setTypeface(enunciado);  
 btnAtrasTerminos.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v) {  
 terminos.dismiss();  
 }  
 });  
  
 terminos.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*));  
 terminos.show();  
 }  
  
 public boolean comprobarNombre(){  
 String nombre;  
 Boolean correcto;  
  
 nombre = edtNombreRegistro.getText().toString();  
 correcto = false;  
  
 if(!nombre.isEmpty() && nombre.length()>4){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el nombre." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean comprobarNombreUsuario(){  
 String nombreUser;  
 Boolean correcto;  
  
 nombreUser = edtNombreUserRegistro.getText().toString();  
 correcto = false;  
  
 if(!nombreUser.isEmpty() && nombreUser.length()>4){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el nombre de usuario." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean comprobarEmail(){  
 String email;  
 Boolean correcto;  
  
 email = edtEmailRegistro.getText().toString();  
 correcto = false;  
  
 if(!email.isEmpty() &&  
 android.util.Patterns.*EMAIL\_ADDRESS*.matcher(email).matches()){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el email." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean comprobarPassword(){  
 String password;  
 String passwordRepite;  
  
 char [] cPassword;  
 char [] cMayusculas;  
 char [] cNumeros;  
  
 Boolean correcto;  
 Boolean mayuscula;  
 Boolean numero;  
  
  
 cNumeros = new char[]{'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'U', 'V', 'W', 'X', 'Y', 'Z'};  
 cMayusculas = new char[]{'0','1','2','3','4','5','6','7','8','9'};  
  
 password = edtPasswordRegistro.getText().toString();  
 passwordRepite = edtPasswordRepiteRegistro.getText().toString();  
 cPassword = password.toCharArray();  
 correcto = false;  
 mayuscula = false;  
 numero = false;  
  
 if(password.equals(passwordRepite)){  
 for(int i = 0; i < cPassword.length; i++){  
 for(int f = 0; f < cNumeros.length; f++){  
 if(cPassword[i]==cNumeros[f]){  
 numero = true;  
 }  
 }  
  
 for(int f = 0; f < cMayusculas.length; f++){  
 if(cPassword[i]==cMayusculas[f]){  
 mayuscula = true;  
 }  
 }  
 }  
  
 if(!password.isEmpty() && password.length()>6 && mayuscula && numero){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en la contraseña." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Las contraseñas no coinciden." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
 return correcto;  
 }  
  
 public boolean comprobarTelefono(){  
 String telefono;  
 Boolean correcto;  
  
 telefono = edtTelefonoRegistro.getText().toString();  
 correcto = false;  
  
 if(!telefono.isEmpty() && telefono.length()==9){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en telefono." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean comprobarDireccion(){  
 String direcion;  
 Boolean correcto;  
  
 direcion = edtDireccionRegistro.getText().toString();  
 correcto = false;  
  
 if(!direcion.isEmpty() && direcion.length()>5){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en la direccion." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean nacimiento(){  
 int ano;  
 Boolean correcto;  
  
 correcto = false;  
  
 ano = Integer.*parseInt*(spAnoRegistro.getSelectedItem().toString());  
  
 if(ano <= 1999){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el nacimiento, mayor de 18 años" , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public void registrarse(View v){  
 if(comprobarNombre() && comprobarNombreUsuario() && comprobarEmail() && comprobarPassword()  
 && comprobarTelefono() && comprobarDireccion() && chkAceptarTerminosRegistro.isChecked()  
 && nacimiento()){  
  
 String nombre = edtNombreRegistro.getText().toString();  
 String nombreusuario = edtNombreUserRegistro.getText().toString();  
 String email = edtEmailRegistro.getText().toString();  
 String password = edtPasswordRegistro.getText().toString();  
 String telefono = edtTelefonoRegistro.getText().toString();  
 String direccion = edtDireccionRegistro.getText().toString();  
 String img = "";  
 String nacimiento = spDiaRegistro.getSelectedItem().toString() +  
 spMesRegistro.getSelectedItem().toString() + spAnoRegistro.getSelectedItem().toString();  
  
 *url* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/registro.php?nombre="+nombre+"&nombreusuario="+nombreusuario+  
 "&email="+email+"&password="+password+"&nacimiento="+nacimiento+"&telefono="+telefono+  
 "&imgperfil="+img+"&direccion="+direccion;  
  
 uploadImage();  
  
 // Ejecutamos el metodo GetMeme  
 new GetPartida().execute();  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "No se ha podido registar." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
 }  
  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/* private class GetPartida extends AsyncTask<Void, Void, Void> {  
  
 // Hashmap for ListView  
 String respuesta;  
  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 // Creating service handler class instance  
 WebRequest webreq = new WebRequest();  
  
 // Making a request to url and getting response  
 respuesta = webreq.makeWebServiceCall(*url*, WebRequest.*GET*);  
  
 Log.*d*("Response: ", ">" + respuesta);  
  
 ParseJSON(respuesta);  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 if(contar.equals("hecho")){  
 Intent i = new Intent(registro.this, login.class);  
 startActivity(i);  
  
 enviarCorreo();  
  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Registro completado, te hemos enviado un correo." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "El email o usuario ya existe." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
 }  
  
 }  
  
 private ArrayList<HashMap<String, String>> ParseJSON(String json) {  
 if (json != null) {  
 try {  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> usuarioList = new ArrayList<>();  
  
 JSONObject jsonObj = new JSONObject(json);  
  
 // Getting JSON Array node  
 JSONArray animales = jsonObj.getJSONArray("registro");  
  
 // looping through All Students  
 for (int i = 0; i < animales.length(); i++) {  
 JSONObject c = animales.getJSONObject(i);  
  
 contar = c.getString("contar");  
  
 // tmp hashmap for single student  
 HashMap<String, String> animal = new HashMap<String, String>();  
  
 // adding each child node to HashMap key => value  
 animal.put("contar", contar);  
  
 // adding student to students list  
 usuarioList.add(animal);  
 }  
  
 return usuarioList;  
 } catch (JSONException e) {  
 e.printStackTrace();  
 return null;  
 }  
 } else {  
 Log.*e*("ServiceHandler", "Couldn't get any data from the url");  
 return null;  
 }  
 }  
  
 public void enviarCorreo(){  
  
 // pongo email y contraseña  
 correo = "adoptalescontacto@gmail.com";  
 password = "Ru83nP1l0m03105";  
  
 // Ponemos las politica  
 StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();  
 StrictMode.*setThreadPolicy*(policy);  
 // Creamos unas propiedades  
 Properties properties = new Properties();  
 // ponemos la conexion  
 properties.put("mail.smtp.auth", "true");  
 properties.put("mail.smtp.starttls.enable", "true");  
 properties.put("mail.smtp.host", "smtp.gmail.com");  
 properties.put("mail.smtp.port", "587");  
  
 // intentamos  
 try {  
 // Inicializamos la sesion  
 session = Session.*getDefaultInstance*(properties, new Authenticator() {  
 @Override  
 // Autenticacion  
 protected PasswordAuthentication getPasswordAuthentication() {  
 return new PasswordAuthentication(correo, password);  
 }  
 });  
  
 // Si la sesion es erronea  
 if (session != null) {  
 // Iniciamos sesion  
 Message message = new MimeMessage(session);  
  
 // Quien lo envia  
 message.setFrom(new InternetAddress(correo));  
  
 // Titulo  
 message.setSubject("Mensaje de contacto");  
  
 // Quien lo recibe  
 message.setRecipients(Message.RecipientType.*TO*, InternetAddress.*parse*(edtEmailRegistro.getText().toString()));  
  
 // Mensaje y su forma  
 message.setContent("Buenas "+edtNombreUserRegistro.getText().toString()  
 +", hemos registrado tu correo "+ edtEmailRegistro.getText().toString()  
 +" en nuestros servidores, esperemos ser de gran ayuda a la hora de encontra su animal.", "text/html; charset=utf-8");  
  
 // Enviamos el mensaje  
 Transport.*send*(message);  
  
 }  
  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Correo enviado.", Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 } catch (AddressException e) {  
 e.printStackTrace();  
 } catch (MessagingException e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
  
  
 private void showFileChooser() {  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 startActivityForResult(Intent.*createChooser*(intent, "Select Imagen"), PICK\_IMAGE\_REQUEST);  
 }  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == PICK\_IMAGE\_REQUEST && resultCode == *RESULT\_OK* && data != null && data.getData() != null) {  
 Uri filePath = data.getData();  
 try {  
//Cómo obtener el mapa de bits de la Galería  
 bitmap = MediaStore.Images.Media.*getBitmap*(getContentResolver(), filePath);  
//Configuración del mapa de bits en ImageView  
 imgSubirAnimal.setImageBitmap(bitmap);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 @Override  
 public void onClick(View v) {  
 if(v == btnAvatar){  
 showFileChooser();  
 }  
 }  
  
 public String getStringImagen(Bitmap bmp){  
 ByteArrayOutputStream baos = new ByteArrayOutputStream();  
 bmp.compress(Bitmap.CompressFormat.*JPEG*, 100, baos);  
 byte[] imageBytes = baos.toByteArray();  
 String encodedImage = Base64.*encodeToString*(imageBytes, Base64.*DEFAULT*);  
 return encodedImage;  
 }  
  
 private void uploadImage(){  
 //Mostrar el diálogo de progreso  
 final ProgressDialog loading = ProgressDialog.*show*(this,"Subiendo...","Espere por favor...",false,false);  
 StringRequest stringRequest = new StringRequest(Request.Method.*POST*, UPLOAD\_URL,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String s) {  
 //Descartar el diálogo de progreso  
 loading.dismiss();  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError volleyError) {  
 //Descartar el diálogo de progreso  
 loading.dismiss();  
  
 //Showing toast  
 Toast.*makeText*(registro.this, volleyError.getMessage().toString(), Toast.*LENGTH\_LONG*).show();  
 }  
 }){  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
  
 bitmap = Bitmap.*createScaledBitmap*(bitmap,700,500, false);  
  
 //Convertir bits a cadena  
 String imagen = getStringImagen(bitmap);  
  
 //Creación de parámetros  
 Map<String,String> params = new Hashtable<String, String>();  
 //Agregando de parámetros  
 params.put(KEY\_IMAGEN, imagen);  
  
 //Parámetros de retorno  
 return params;  
 }  
 };  
 //Creación de una cola de solicitudes  
 RequestQueue requestQueue = Volley.*newRequestQueue*(this);  
 //Agregar solicitud a la cola  
 requestQueue.add(stringRequest);  
 }  
  
}

DI14. Splashscreen.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.app.Activity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
  
*/\*\*  
 \* Created by Ruben Palomo on 16/03/2018.  
 \*/*public class splashscreen extends Activity {  
  
 public void onCreate(Bundle savedInstanceState){  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*splashscreen*);  
  
 // Tiempo de delay de la activity  
 new Handler().postDelayed(new Runnable(){  
 @Override  
 public void run() {  
 // Creamos un intento a la paginaPrincipal  
 Intent i = new Intent(splashscreen.this, paginaPrincipal.class);  
 // Empezamos la actividad  
 startActivity(i);  
 // Tiempo de espera  
 }  
 }, 1500);  
 }  
}

DI15. Subiranimal.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Bitmap;  
import android.graphics.Typeface;  
import android.net.Uri;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.provider.MediaStore;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.util.Base64;  
import android.util.Log;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.RadioButton;  
import android.widget.Spinner;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.io.ByteArrayOutputStream;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Hashtable;  
import java.util.Map;  
  
import com.android.volley.AuthFailureError;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.StringRequest;  
import com.android.volley.toolbox.Volley;  
  
public class subiranimal extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener, View.OnClickListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // TextView del refugio  
 TextView enunciadoSubir, txtNombreAnimalSubir, txtTipoAnimalSubir, txtRazaAnimalSubir, txtDescripcionAnimalSubir, txtEdadAnimalSubir  
 , txtGeneroAnimalSubir, txtVacunaAnimalSubir, txtChipAnimalSubir;  
  
 // Button del refugio  
 Button btnSubirAnimal, btnAvatar;  
  
 Spinner spTipoAnimalSubir;  
  
 // EditText de contacto  
 EditText edtNombreAnimalSubir, edtRazaAnimalSubir, edtDescripcionAnimalSubir, edtEdadAnimalSubir;  
  
 RadioButton rbGeneroMAnimalSubir, rbGeneroHAnimalSubir, rbVacunaSAnimalSubir, rbVacunaNAnimalSubir,rbChipSAnimalSubir  
 ,rbChipNAnimalSubir;  
 ImageView imgSubirAnimal;  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 // URL to get contacts JSON  
 private static String *url*;  
 String result;  
 String iduser;  
  
 // Subir imagen  
 private Bitmap bitmap;  
 private int PICK\_IMAGE\_REQUEST = 1;  
 private String UPLOAD\_URL ="https://www.losnaranjosdam.online/2dam/ruben/adoptales/subirFoto.php";  
 private String KEY\_IMAGEN = "foto";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_subiranimal*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave version  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
  
 // Si esta logueado  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Cogemos los extras que nos pasan en la anterio activity  
 Bundle extras = getIntent().getExtras();  
  
 // Si los extras no son nulos  
 if (extras != null) {  
 // Guardo los datos que tengan como clave ano  
 iduser = extras.getString("iduser");  
 } else {  
 // Se queda vacio  
 iduser = "";  
 }  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 enunciadoSubir = findViewById(R.id.*enunciadoSubir*);  
 txtNombreAnimalSubir = findViewById(R.id.*txtNombreAnimalSubir*);  
 txtTipoAnimalSubir = findViewById(R.id.*txtTipoAnimalSubir*);  
 txtRazaAnimalSubir = findViewById(R.id.*txtRazaAnimalSubir*);  
 txtDescripcionAnimalSubir = findViewById(R.id.*txtDescripcionAnimalSubir*);  
 txtEdadAnimalSubir = findViewById(R.id.*txtEdadAnimalSubir*);  
 txtGeneroAnimalSubir= findViewById(R.id.*txtGeneroAnimalSubir*);  
 txtVacunaAnimalSubir = findViewById(R.id.*txtVacunaAnimalSubir*);  
 txtChipAnimalSubir = findViewById(R.id.*txtChipAnimalSubir*);  
 rbGeneroMAnimalSubir = findViewById(R.id.*rbGeneroMAnimalSubir*);  
 rbGeneroHAnimalSubir = findViewById(R.id.*rbGeneroHAnimalSubir*);  
 rbVacunaSAnimalSubir = findViewById(R.id.*rbVacunaSAnimalSubir*);  
 rbVacunaNAnimalSubir = findViewById(R.id.*rbVacunaNAnimalSubir*);  
 rbChipSAnimalSubir = findViewById(R.id.*rbChipSAnimalSubir*);  
 rbChipNAnimalSubir = findViewById(R.id.*rbChipNAnimalSubir*);  
 btnSubirAnimal = findViewById(R.id.*btnSubirAnimal*);  
 btnAvatar = findViewById(R.id.*btnAvatar*);  
 spTipoAnimalSubir = findViewById(R.id.*spTipoAnimalSubir*);  
 edtNombreAnimalSubir = findViewById(R.id.*edtNombreAnimalSubir*);  
 edtRazaAnimalSubir = findViewById(R.id.*edtRazaAnimalSubir*);  
 edtDescripcionAnimalSubir = findViewById(R.id.*edtDescripcionAnimalSubir*);  
 edtEdadAnimalSubir = findViewById(R.id.*edtEdadAnimalSubir*);  
 imgSubirAnimal = findViewById(R.id.*imgSubirAnimal*);  
  
 // Ponemos las fuentes  
 enunciadoSubir.setTypeface(enunciado);  
 txtNombreAnimalSubir.setTypeface(texto);  
 txtTipoAnimalSubir.setTypeface(texto);  
 txtRazaAnimalSubir.setTypeface(texto);  
 txtDescripcionAnimalSubir.setTypeface(texto);  
 txtEdadAnimalSubir.setTypeface(texto);  
 txtGeneroAnimalSubir.setTypeface(texto);  
 txtVacunaAnimalSubir.setTypeface(texto);  
 txtChipAnimalSubir.setTypeface(texto);  
 rbGeneroMAnimalSubir.setTypeface(texto);  
 rbGeneroHAnimalSubir.setTypeface(texto);  
 rbVacunaSAnimalSubir.setTypeface(texto);  
 rbVacunaNAnimalSubir.setTypeface(texto);  
 rbChipSAnimalSubir.setTypeface(texto);  
 rbChipNAnimalSubir.setTypeface(texto);  
 btnSubirAnimal.setTypeface(texto);  
 btnAvatar.setTypeface(texto);  
  
 // Adaptamos los datos a un spinner  
 ArrayAdapter<CharSequence> adapter = ArrayAdapter.*createFromResource*(this, R.array.*tipoanimal*, android.R.layout.*simple\_spinner\_item*);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 spTipoAnimalSubir.setAdapter(adapter);  
  
 btnAvatar.setOnClickListener(this);  
 }  
  
 public void subirNuevoAnimal(View v){  
 if(comprobarNombre() && comprobarEdad() && comprobarRaza() && comprobarDescripcion() && comprobarImg()  
 && comprobarChip() && comprobarGenero() && comprobarVacuna()){  
 String nombre = edtNombreAnimalSubir.getText().toString();  
 String tipo = spTipoAnimalSubir.getSelectedItem().toString();  
 String raza = edtRazaAnimalSubir.getText().toString();  
 String descripcion = edtDescripcionAnimalSubir.getText().toString();  
 String edad = edtEdadAnimalSubir.getText().toString();  
 String vacuna = "0";  
 String chip = "0";  
 String genero = "Hembra";  
  
 if(rbVacunaSAnimalSubir.isChecked()){  
 vacuna = "1";  
 }  
 if(rbChipSAnimalSubir.isChecked()){  
 chip = "1";  
 }  
 if(rbGeneroMAnimalSubir.isChecked()){  
 genero = "Macho";  
 }  
  
  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 iduser , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
  
 if (!iduser.equals("0")) {  
 *url* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/subirAnimal.php?nombre="+nombre+"&tipo="+tipo+  
 "&raza="+raza+"&descripcion="+descripcion+"&imganimal=img1&edad="+edad+  
 "&iduser="+iduser+"&genero="+genero+"&chip="+chip+"&vacuna="+vacuna;  
  
 uploadImage();  
  
 new GetPartida().execute();  
 }  
 }  
 }  
  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/* private class GetPartida extends AsyncTask<Void, Void, Void> {  
  
 // Hashmap for ListView  
 String respuesta;  
  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 // Creating service handler class instance  
 WebRequest webreq = new WebRequest();  
  
 // Making a request to url and getting response  
 respuesta = webreq.makeWebServiceCall(*url*, WebRequest.*GET*);  
  
 Log.*d*("Response: ", ">" + respuesta);  
  
 ParseJSON(respuesta);  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void resulte) {  
 super.onPostExecute(resulte);  
  
 if(result.equals("hecho")){  
 Intent i = new Intent(subiranimal.this, cuenta.class);  
 startActivity(i);  
  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Registro completado, te hemos enviado un correo." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "El email o usuario ya existe." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
 }  
  
 }  
  
 private ArrayList<HashMap<String, String>> ParseJSON(String json) {  
 if (json != null) {  
 try {  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> usuarioList = new ArrayList<>();  
  
 JSONObject jsonObj = new JSONObject(json);  
  
 // Getting JSON Array node  
 JSONArray animales = jsonObj.getJSONArray("subir");  
  
 // looping through All Students  
 for (int i = 0; i < animales.length(); i++) {  
 JSONObject c = animales.getJSONObject(i);  
  
 result = c.getString("resu");  
  
 // tmp hashmap for single student  
 HashMap<String, String> animal = new HashMap<String, String>();  
  
 // adding each child node to HashMap key => value  
 animal.put("resu", result);  
  
 // adding student to students list  
 usuarioList.add(animal);  
 }  
  
 return usuarioList;  
 } catch (JSONException e) {  
 e.printStackTrace();  
 return null;  
 }  
 } else {  
 Log.*e*("ServiceHandler", "Couldn't get any data from the url");  
 return null;  
 }  
 }  
  
 public boolean comprobarGenero(){  
 Boolean correcto;  
  
 correcto = false;  
  
 if(rbGeneroHAnimalSubir.isChecked() || rbGeneroMAnimalSubir.isChecked()){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el genero." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean comprobarChip(){  
 Boolean correcto;  
  
 correcto = false;  
  
 if(rbChipNAnimalSubir.isChecked() || rbChipSAnimalSubir.isChecked()){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el chip." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean comprobarVacuna(){  
 Boolean correcto;  
  
 correcto = false;  
  
 if(rbVacunaNAnimalSubir.isChecked() || rbVacunaSAnimalSubir.isChecked()){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en la vacuna." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean comprobarNombre(){  
  
 String nombre;  
 Boolean correcto;  
  
 nombre = edtNombreAnimalSubir.getText().toString();  
 correcto = false;  
  
 if(!nombre.isEmpty() && nombre.length()>4){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en el nombre." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
  
 }  
  
 public boolean comprobarEdad(){  
  
 String edad;  
 int edadNumero;  
 Boolean correcto = false;  
  
 try {  
 edad = edtEdadAnimalSubir.getText().toString();  
 edadNumero = Integer.*parseInt*(edtEdadAnimalSubir.getText().toString());  
 correcto = false;  
  
 if (!edad.isEmpty() && edadNumero > 0 && edadNumero < 60) {  
 correcto = true;  
 } else {  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en la edad.", Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 }catch (NumberFormatException e){  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en la edad.", Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
  
 }  
  
 public boolean comprobarDescripcion(){  
 String descripcion;  
 Boolean correcto;  
  
 descripcion = edtDescripcionAnimalSubir.getText().toString();  
 correcto = false;  
  
 if(!descripcion.isEmpty() && descripcion.length()>10){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Necesitamos una descripcion un poco mas larga." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
 }  
  
 public boolean comprobarRaza(){  
  
 String raza;  
 Boolean correcto;  
  
 raza = edtRazaAnimalSubir.getText().toString();  
 correcto = false;  
  
 if(!raza.isEmpty() && raza.length()>4){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Error en la raza." , Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
 }  
  
 return correcto;  
  
 }  
  
 public boolean comprobarImg(){  
  
 Boolean correcto;  
  
 correcto = true;  
*/\*\*  
 if(!imgSubirAnimal.getDrawable().equals(R.drawable.icono)){  
 correcto = true;  
 }else{  
 Toast toast1 = Toast.makeText(getApplicationContext(),  
 "No has puesto una imagen." , Toast.LENGTH\_SHORT);  
  
 toast1.show();  
 }\*/* return correcto;  
  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 Intent i;  
  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos la variable el entero con clave version  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
  
 private void showFileChooser() {  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 startActivityForResult(Intent.*createChooser*(intent, "Select Imagen"), PICK\_IMAGE\_REQUEST);  
 }  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == PICK\_IMAGE\_REQUEST && resultCode == *RESULT\_OK* && data != null && data.getData() != null) {  
 Uri filePath = data.getData();  
 try {  
//Cómo obtener el mapa de bits de la Galería  
 bitmap = MediaStore.Images.Media.*getBitmap*(getContentResolver(), filePath);  
//Configuración del mapa de bits en ImageView  
 imgSubirAnimal.setImageBitmap(bitmap);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 @Override  
 public void onClick(View v) {  
 if(v == btnAvatar){  
 showFileChooser();  
 }  
 }  
  
 public String getStringImagen(Bitmap bmp){  
 ByteArrayOutputStream baos = new ByteArrayOutputStream();  
 bmp.compress(Bitmap.CompressFormat.*JPEG*, 100, baos);  
 byte[] imageBytes = baos.toByteArray();  
 String encodedImage = Base64.*encodeToString*(imageBytes, Base64.*DEFAULT*);  
 return encodedImage;  
 }  
  
 private void uploadImage(){  
 //Mostrar el diálogo de progreso  
 final ProgressDialog loading = ProgressDialog.*show*(this,"Subiendo...","Espere por favor...",false,false);  
 StringRequest stringRequest = new StringRequest(Request.Method.*POST*, UPLOAD\_URL,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String s) {  
 //Descartar el diálogo de progreso  
 loading.dismiss();  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError volleyError) {  
 //Descartar el diálogo de progreso  
 loading.dismiss();  
  
 //Showing toast  
 Toast.*makeText*(subiranimal.this, volleyError.getMessage().toString(), Toast.*LENGTH\_LONG*).show();  
 }  
 }){  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
  
 bitmap = Bitmap.*createScaledBitmap*(bitmap,1000,800, false);  
  
 //Convertir bits a cadena  
 String imagen = getStringImagen(bitmap);  
  
 //Creación de parámetros  
 Map<String,String> params = new Hashtable<String, String>();  
 //Agregando de parámetros  
 params.put(KEY\_IMAGEN, imagen);  
 //Agregando de parámetros  
 params.put("id", iduser+".png");  
  
 //Parámetros de retorno  
 return params;  
 }  
 };  
 //Creación de una cola de solicitudes  
 RequestQueue requestQueue = Volley.*newRequestQueue*(this);  
 //Agregar solicitud a la cola  
 requestQueue.add(stringRequest);  
 }  
  
}

DI16. Tuanimal.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Typeface;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.support.annotation.NonNull;  
import android.support.design.widget.BottomNavigationView;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.util.Log;  
import android.view.Display;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.squareup.picasso.Picasso;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
public class tuanimal extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
  
 // Fuentes  
 Typeface enunciado, texto;  
  
 // TextView del refugio  
 TextView txtNombreAnimal, txtNumeroAnimal, txtEmailAnimal, txtDescripcionAnimal, txtChipTu, txtVacunaTu, txtGeneroTu  
 , txtEdadTu, txtTipoAnimal, txtInfoAnimal, txtInformacion;  
  
 // ImageView del refugio  
 ImageView imgAnimal;  
  
 // Registro  
 SharedPreferences mPrefs;  
 SharedPreferences.Editor myEditor;  
  
 // Datos  
 String id, nombre, raza, tipo, descripcion, imganimal, edad, iduser, email, telefono;  
 String chip;  
 String vacuna;  
 String genero;  
  
 LinearLayout ly1, ly2, ly3, ly4;  
  
 // PHP  
 private static String *url*;  
 private static final String *TAG\_RESPUESTAINFO* = "respuesta";  
 private static final String *TAG\_RESPUESTA* = "resp";  
 String respuesta;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_tuanimal*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 // Creamos un registro con esa id  
 mPrefs = getSharedPreferences("IDvalue", 0);  
 // Creamos un editor del registro  
 myEditor = mPrefs.edit();  
 // Actualizmos la variable el entero con clave logueado  
 Boolean logeado = mPrefs.getBoolean("logeado", false);  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.*navigation*);  
 navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);  
  
 // Creamos el menu y le damos el funcionamiento  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
  
 ly1 = findViewById(R.id.*ly1*);  
 ly2 = findViewById(R.id.*ly2*);  
 ly3 = findViewById(R.id.*ly3*);  
 ly4 = findViewById(R.id.*ly4*);  
  
 Display display = getWindowManager().getDefaultDisplay();  
 int width = display.getWidth() / 4; // obsoleto (deprecated)  
  
 ly1.getLayoutParams().width=width;  
 ly2.getLayoutParams().width=width;  
 ly3.getLayoutParams().width=width;  
 ly4.getLayoutParams().width=width;  
  
 // Comprobamos que esta logueado  
 // Se pone un menu o otro  
 if(logeado == true) {  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_conregistro\_drawer*);  
 }else{  
 navigationView.inflateMenu(R.menu.*activity\_pagina\_sinregistro\_drawer*);  
 }  
  
 // Inicializamos las fuentes  
 enunciado = Typeface.*createFromAsset*(getAssets(),"fonts/Happy-Camper-Regular.ttf");  
 texto = Typeface.*createFromAsset*(getAssets(),"fonts/Crocodile Feet DEMO.otf");  
  
 // Inicializamos los widgets  
 txtNombreAnimal = findViewById(R.id.*txtNombreAnimal*);  
 txtNumeroAnimal = findViewById(R.id.*txtNumeroAnimal*);  
 txtEmailAnimal = findViewById(R.id.*txtEmailAnimal*);  
 txtDescripcionAnimal = findViewById(R.id.*txtDescripcionAnimal*);  
 txtChipTu = findViewById(R.id.*txtChipTu*);  
 txtVacunaTu = findViewById(R.id.*txtVacunaTu*);  
 txtGeneroTu = findViewById(R.id.*txtGeneroTu*);  
 txtEdadTu = findViewById(R.id.*txtEdadTu*);  
 txtTipoAnimal = findViewById(R.id.*txtTipoAnimal*);  
 txtInfoAnimal = findViewById(R.id.*txtInfoAnimal*);  
 txtInformacion = findViewById(R.id.*txtInformacion*);  
 imgAnimal = findViewById(R.id.*imgAnimal*);  
  
 // Ponemos las fuentes  
 txtNombreAnimal.setTypeface(enunciado);  
 txtChipTu.setTypeface(texto);  
 txtVacunaTu.setTypeface(texto);  
 txtGeneroTu.setTypeface(texto);  
 txtEdadTu.setTypeface(texto);  
 txtInformacion.setTypeface(texto);  
 txtInfoAnimal.setTypeface(texto);  
  
 // Cogemos los extras que nos pasan en la anterio activity  
 Bundle extras = getIntent().getExtras();  
  
 // Si los extras no son nulos  
 if (extras != null) {  
 // Guardo los datos con su clave  
 id = extras.getString("id");  
 nombre = extras.getString("nombre");  
 raza = extras.getString("raza");  
 tipo = extras.getString("tipo");  
 descripcion = extras.getString("descripcion");  
 imganimal = extras.getString("imganimal");  
 edad = extras.getString("edad");  
 iduser = extras.getString("iduser");  
 chip = extras.getString("chip");  
 vacuna = extras.getString("vacuna");  
 genero = extras.getString("genero");  
 telefono = extras.getString("telefono");  
 email = extras.getString("email");  
 } else {  
 // Se queda vacio  
 id = "";  
 nombre = "";  
 raza = "";  
 tipo = "";  
 descripcion = "";  
 imganimal = "";  
 edad = "";  
 iduser = "";  
 }  
  
 // Ponemos los textos  
 txtNombreAnimal.setText(nombre);  
 txtDescripcionAnimal.setText(descripcion);  
 txtTipoAnimal.setText("-"+tipo+", "+ raza);  
 txtEmailAnimal.setText("-"+ email);  
 txtNumeroAnimal.setText("-"+ telefono);  
 txtEdadTu.setText(edad);  
 txtDescripcionAnimal.setText("-"+descripcion);  
 txtChipTu.setText(chip);  
 txtVacunaTu.setText(vacuna);  
 txtGeneroTu.setText(genero);  
  
 // Ponemos la imagen de perfil  
 Picasso.*with*(this)  
 .load("https://www.losnaranjosdam.online/2dam/ruben/adoptales/fotos/"+ imganimal)  
 .error(R.mipmap.*ic\_launcher*)  
 .into(imgAnimal);  
  
 }  
  
 */\*\*  
 \* Borramos un animal  
 \*/* public void borrarAnimal(){  
  
 *url* = "https://www.losnaranjosdam.online/2dam/ruben/adoptales/borrarAnimal.php?id="+ id;  
  
 // Ejecutamos el metodo GetMeme  
 new GetDatosDueno().execute();  
 }  
  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/* private class GetDatosDueno extends AsyncTask<Void, Void, Void> {  
  
 // Hashmap for ListView  
 ArrayList<HashMap<String, String>> userList;  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 // Creating service handler class instance  
 WebRequest webreq = new WebRequest();  
  
 // Making a request to url and getting response  
 String json = webreq.makeWebServiceCall(*url*, WebRequest.*GET*);  
  
 Log.*d*("Response: ", "> " + respuesta);  
  
 // Llamamos al metodo que te convierte el json  
 ParseJSON(json);  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 // Si la respuesta es hecho  
 if(respuesta.equals("hecho")){  
 // Mensaje de informacion  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "Se ha borrado la mascota" , Toast.*LENGTH\_SHORT*);  
 toast1.show();  
  
 // Nos movemos a otra activity  
 Intent i = new Intent(tuanimal.this, cuenta.class);  
 startActivity(i);  
 }else{  
 // Mensaje de error  
 Toast toast1 = Toast.*makeText*(getApplicationContext(),  
 "No se ha borrado la mascota" , Toast.*LENGTH\_SHORT*);  
 toast1.show();  
 }  
 }  
 }  
  
 private void ParseJSON(String json) {  
 // Si el json es nulo  
 if (json != null) {  
 try {  
 // Hacemos un objeto json con el json pasado por parametro  
 JSONObject jsonObj = new JSONObject(json);  
  
 // Cogemos el nodo del JSON y creamos un array  
 JSONArray usuario = jsonObj.getJSONArray(*TAG\_RESPUESTAINFO*);  
  
 // Recorremos todos los usuarios  
 for (int i = 0; i < usuario.length(); i++) {  
 // Cogemos el objeto json en esa posivion  
 JSONObject c = usuario.getJSONObject(i);  
  
 // Ponemos la respuesta, con esa clave  
 respuesta = c.getString(*TAG\_RESPUESTA*);  
 }  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 } else {  
 Log.*e*("ServiceHandler", "Couldn't get any data from the url");  
 }  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 // Creamos el menu  
 getMenuInflater().inflate(R.menu.*pagina\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 // Creamos un intento  
 Intent i;  
  
 // Depende de lo que pulse hace una opcion  
 if (id == R.id.*action\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_info*){  
 i = new Intent(this, info.class);  
 startActivity(i);  
 return true;  
 } else if (id == R.id.*action\_ayuda*){  
 i = new Intent(this, ayuda.class);  
 startActivity(i);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 // Creamos un intento  
 Intent i;  
  
 // Depende de lo que pulse hace una opcion  
 if (id == R.id.*nav\_inicio*) {  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_inicioSesion*) {  
 i = new Intent(this, login.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_registro*) {  
 i = new Intent(this, registro.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_refugios*) {  
 i = new Intent(this, refugios.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cuenta*) {  
 i = new Intent(this, cuenta.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_compartir*) {  
 i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, "App para dar vida a los animales.");  
 startActivity(i);  
 } else if (id == R.id.*nav\_contacto*) {  
 i = new Intent(this, contacto.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_info*) {  
 i = new Intent(this, info.class);  
 startActivity(i);  
 } else if (id == R.id.*nav\_cerrarsesion*){  
 // Actualizmos el registro  
 myEditor.putBoolean("logeado", false);  
 myEditor.putString("emailuser", "");  
 myEditor.putString("password", "");  
 // Hacemos persistentes los datos  
 myEditor.commit();  
  
 i = new Intent(this, paginaPrincipal.class);  
 startActivity(i);  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
  
  
 public BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener  
 = new BottomNavigationView.OnNavigationItemSelectedListener() {  
  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*navigation\_llamar*:  
 borrarAnimal();  
 return true;  
 case R.id.*navigation\_maps*:  
 borrarAnimal();  
 return true;  
 }  
 return false;  
 }  
 };  
}

DI17. UsuarioSQLiteHepler.

package com.adoptales.rubenpalomo.adoptales;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
*/\*\*  
 \* Created by Ruben Palomo on 13/12/2017.  
 \*/*public class UsuarioSQLiteHelper extends SQLiteOpenHelper {  
  
 // SQL para crea la tabla  
 String crearTabla1 = "CREATE TABLE animales(id INTEGER PRIMARY KEY ,nombre TEXT, tipo TEXT, raza TEXT ,descripcion TEXT ,imganimal TEXT ,edad TEXT ,iduser TEXT ,vacunado TEXT ,chip TEXT ,genero TEXT)";  
 String crearTabla2 = "CREATE TABLE refugios(id INTEGER PRIMARY KEY ,nombrerefugio TEXT, nombreadmin TEXT, email TEXT, telefono TEXT, imgperfil TEXT, direccion TEXT)";  
  
 */\*\*  
 \** ***@param*** *context  
 \** ***@param*** *name  
 \** ***@param*** *factory  
 \** ***@param*** *version  
 \*/* public UsuarioSQLiteHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {  
 super(context, name, factory, version);  
 }  
  
 */\*\*  
 \* Si se ejecuta se crea la tabla  
 \** ***@param*** *sqLiteDatabase  
 \*/* @Override  
 public void onCreate(SQLiteDatabase sqLiteDatabase) {  
  
 sqLiteDatabase.execSQL(crearTabla1);  
 sqLiteDatabase.execSQL(crearTabla2);  
 }  
  
 */\*\*  
 \* Cuando se actualiza la version se actualiza la base de datos  
 \** ***@param*** *sqLiteDatabase  
 \** ***@param*** *i  
 \** ***@param*** *i1  
 \*/* @Override  
 public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {  
 // Borramos la tabla  
 sqLiteDatabase.execSQL("Delete from animales");  
 sqLiteDatabase.execSQL("Delete from refugios");  
  
 // La quitamos si existe  
 sqLiteDatabase.execSQL("DROP TABLE IF EXISTS animales");  
 sqLiteDatabase.execSQL("DROP TABLE IF EXISTS refugios");  
  
 // Creamos la nueva table  
 sqLiteDatabase.execSQL(crearTabla1);  
 sqLiteDatabase.execSQL(crearTabla2);  
 }  
}

DI18. WebRequest.

package com.adoptales.rubenpalomo.adoptales;  
  
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.InputStreamReader;  
import java.io.OutputStream;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.net.URLEncoder;  
import java.util.HashMap;  
import java.util.Map;  
  
import javax.net.ssl.HttpsURLConnection;  
  
*/\*\*  
 \* Created by Ruben Palomo on 13/12/2017.  
 \*/*public class WebRequest {  
  
 static String *response* = null;  
 public final static int *GET* = 1;  
 public final static int *POST* = 2;  
  
 //Constructor with no parameter  
 public WebRequest() {  
  
 }  
  
 */\*\*  
 \* Making web service call  
 \*  
 \** ***@url*** *- url to make request  
 \** ***@requestmethod*** *- http request method  
 \*/* public String makeWebServiceCall(String url, int requestmethod) {  
 return this.makeWebServiceCall(url, requestmethod, null);  
 }  
  
 */\*\*  
 \* Making service call  
 \*  
 \** ***@url*** *- url to make request  
 \** ***@requestmethod*** *- http request method  
 \** ***@params*** *- http request params  
 \*/* public String makeWebServiceCall(String urladdress, int requestmethod,  
 HashMap<String, String> params) {  
 URL url;  
 String response = "";  
 try {  
 url = new URL(urladdress);  
  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 conn.setReadTimeout(15000);  
 conn.setConnectTimeout(15000);  
 conn.setDoInput(true);  
  
 if (requestmethod == *POST*) {  
 conn.setRequestMethod("POST");  
 } else if (requestmethod == *GET*) {  
 conn.setRequestMethod("GET");  
 }  
  
 if (params != null) {  
 OutputStream os = conn.getOutputStream();  
 BufferedWriter writer = new BufferedWriter(  
 new OutputStreamWriter(os, "UTF-8"));  
  
 StringBuilder result = new StringBuilder();  
 boolean first = true;  
 for (Map.Entry<String, String> entry : params.entrySet()) {  
 if (first)  
 first = false;  
 else  
 result.append("&");  
  
 result.append(URLEncoder.*encode*(entry.getKey(), "UTF-8"));  
 result.append("=");  
 result.append(URLEncoder.*encode*(entry.getValue(), "UTF-8"));  
 }  
  
 writer.write(result.toString());  
  
 writer.flush();  
 writer.close();  
 os.close();  
 }  
  
 int responseCode = conn.getResponseCode();  
  
 if (responseCode == HttpsURLConnection.*HTTP\_OK*) {  
 String line;  
 BufferedReader br = new BufferedReader(new InputStreamReader(conn.getInputStream()));  
 while ((line = br.readLine()) != null) {  
 response += line;  
 }  
 } else {  
 response = "";  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return response;  
 }  
  
}

DI19. Activity\_animal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_animal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial" />  
  
</android.support.v4.widget.DrawerLayout>

DI20. Activity\_ayuda.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_ayuda"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial"/>  
  
</android.support.v4.widget.DrawerLayout>

DI21. Activity\_contacto.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_contacto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial"/>  
  
</android.support.v4.widget.DrawerLayout>

DI22. Activity\_cuenta.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_cuenta"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial"/>  
   
</android.support.v4.widget.DrawerLayout>

DI23. Activity\_info.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_info"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial" />  
  
</android.support.v4.widget.DrawerLayout>

DI24. Activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_login"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial"/>  
  
</android.support.v4.widget.DrawerLayout>

DI25. Activity\_pagina\_principal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_pagina\_principal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial" />  
  
</android.support.v4.widget.DrawerLayout>

DI26. Activity\_regufio.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_refugio"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial" />  
  
</android.support.v4.widget.DrawerLayout>

DI27. Activity\_refugios.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_refugios"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial" />  
  
</android.support.v4.widget.DrawerLayout>

DI28. Activity\_registro.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_registro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial"/>  
  
</android.support.v4.widget.DrawerLayout>

DI29. Activity\_subiranimal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_subiranimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial" />  
  
</android.support.v4.widget.DrawerLayout>

DI30. Activity\_tuanimal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_tuanimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_oficial" />  
  
</android.support.v4.widget.DrawerLayout>

DI31. App\_bar\_animal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.animal">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include layout="@layout/content\_animal" />  
</android.support.design.widget.CoordinatorLayout>

DI32. App\_bar\_ayuda.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.ayuda">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include  
 layout="@layout/content\_ayuda"/>  
</android.support.design.widget.CoordinatorLayout>

DI33. App\_bar\_contacto.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.contacto">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include  
 layout="@layout/content\_contacto"/>  
</android.support.design.widget.CoordinatorLayout>

DI34. App\_bar\_cuenta.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.cuenta">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <android.support.design.widget.FloatingActionButton  
 android:id="@+id/fab"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom|end"  
 app:backgroundTint="#2a867d"  
 android:layout\_margin="@dimen/fab\_margin"  
 app:srcCompat="@drawable/anadir" />  
  
  
 <include layout="@layout/content\_cuenta" />  
</android.support.design.widget.CoordinatorLayout>

DI35. App\_bar\_info.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.info">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include  
 layout="@layout/content\_info"/>  
</android.support.design.widget.CoordinatorLayout>

DI36. App\_bar\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.login">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include  
 layout="@layout/content\_login"/>  
</android.support.design.widget.CoordinatorLayout>

DI37. App\_bar\_pagina\_principal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.paginaPrincipal">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include layout="@layout/content\_pagina\_principal" />  
</android.support.design.widget.CoordinatorLayout>

DI38. App\_bar\_refugio.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.refugio">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include layout="@layout/content\_refugio" />  
</android.support.design.widget.CoordinatorLayout>

DI39. App\_bar\_refugios.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.refugios">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include layout="@layout/content\_refugios" />  
</android.support.design.widget.CoordinatorLayout>

DI40. App\_bar\_registro.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.registro">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include  
 layout="@layout/content\_registro"/>  
</android.support.design.widget.CoordinatorLayout>

DI41. App\_bar\_subiranimal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.subiranimal">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include layout="@layout/content\_subiranimal" />  
</android.support.design.widget.CoordinatorLayout>

DI42. App\_bar\_tuanimal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.tuanimal">  
  
 <android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 </android.support.design.widget.AppBarLayout>  
  
 <include layout="@layout/content\_tuanimal" />  
</android.support.design.widget.CoordinatorLayout>

DI43. Contenido\_flitros.xml.

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="350dp"  
 android:layout\_height="475dp"  
 android:layout\_gravity="center"  
 android:orientation="vertical"  
 android:background="@drawable/datoscuenta">  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@drawable/datoscuenta"  
 android:padding="10dp">  
  
 <TextView  
 android:id="@+id/enunciadoBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Buscar animal"  
 android:layout\_marginBottom="20dp"  
 android:textSize="50dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="#151414"/>  
  
 <TextView  
 android:id="@+id/txtNombreAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Nombre mascota: "/>  
  
 <EditText  
 android:id="@+id/edtNombreAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Su nombre..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPersonName"/>  
  
 <TextView  
 android:id="@+id/txtTipoAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Tipo de la mascota: "/>  
  
 <Spinner  
 android:id="@+id/spTipoAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPersonName"/>  
  
 <TextView  
 android:id="@+id/txtRazaAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Raza de la mascota: "/>  
  
 <EditText  
 android:id="@+id/edtRazaAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Su raza..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPersonName"/>  
  
 <TextView  
 android:id="@+id/txtEdadAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Edad de la mascota: "/>  
  
 <EditText  
 android:id="@+id/edtEdadAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="2..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="number"/>  
  
 <TextView  
 android:id="@+id/txtGeneroAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Genero de la mascota: "/>  
  
 <RadioGroup  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginBottom="10dp">  
 <RadioButton  
 android:id="@+id/rbGeneroMAnimalBuscar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_marginRight="10dp"  
 android:textSize="18dp"  
 android:text="Macho" />  
 <RadioButton  
 android:id="@+id/rbGeneroHAnimalBuscar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:textSize="18dp"  
 android:text="Hembra" />  
 </RadioGroup>  
  
 <TextView  
 android:id="@+id/txtVacunaAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="¿Esta vacunado/a?"/>  
  
 <RadioGroup  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginBottom="10dp">  
 <RadioButton  
 android:id="@+id/rbVacunaSAnimalBuscar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_marginRight="10dp"  
 android:textSize="18dp"  
 android:text="Si" />  
 <RadioButton  
 android:id="@+id/rbVacunaNAnimalBuscar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:textSize="18dp"  
 android:text="No" />  
 </RadioGroup>  
  
 <TextView  
 android:id="@+id/txtChipAnimalBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="¿Tiene chip?"/>  
  
 <RadioGroup  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginBottom="10dp">  
 <RadioButton  
 android:id="@+id/rbChipSAnimalBuscar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_marginRight="10dp"  
 android:textSize="18dp"  
 android:text="Si" />  
 <RadioButton  
 android:id="@+id/rbChipNAnimalBuscar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:textSize="18dp"  
 android:text="No" />  
 </RadioGroup>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:layout\_gravity="center"  
 android:gravity="center">  
  
 <Button  
 android:id="@+id/btnBuscarAnimal"  
 android:layout\_width="125dp"  
 android:layout\_height="wrap\_content"  
 android:text="buscar animal"  
 android:background="@drawable/redondeadobutton"  
 android:textSize="20dp"  
 android:textColor="#ffffff"  
 android:layout\_marginRight="10dp"/>  
  
 <Button  
 android:id="@+id/btnAtrasFiltros"  
 android:layout\_width="125dp"  
 android:layout\_height="wrap\_content"  
 android:text="Atras"  
 android:background="@drawable/redondeadobutton"  
 android:textSize="20dp"  
 android:textColor="#ffffff" />  
 </LinearLayout>  
  
 </LinearLayout>  
  
 </ScrollView>  
</LinearLayout>

DI44. Contenido\_terminos.xml.

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="350dp"  
 android:layout\_height="475dp"  
 android:layout\_gravity="center"  
 android:orientation="vertical"  
 android:background="@drawable/datoscuenta">  
  
 <TextView  
 android:id="@+id/txtEnunciadoTerminos"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Terminos y Condiciones"  
 tools:layout\_editor\_absoluteX="105dp"  
 tools:layout\_editor\_absoluteY="118dp"  
 android:textSize="40dp"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:textColor="#000"/>  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/txtTerminos"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Los siguientes términos y condiciones (los Términos y Condiciones) rigen el uso que usted le dé a este sitio web y a cualquiera de los contenidos disponibles por o a través de este sitio web, incluyendo cualquier contenido derivado del mismo (el Sitio Web). Time Inc. (Time Inc. o nosotros) ha puesto a su disposición el Sitio Web. Podemos cambiar los Términos y Condiciones de vez en cuando, en cualquier momento sin ninguna notificación, sólo publicando los cambios en el Sitio Web. AL USAR EL SITIO WEB, USTED ACEPTA Y ESTÉ DE ACUERDO CON ESTOS TÉRMINOS Y CONDICIONES EN LO QUE SE REFIERE A SU USO DEL SITIO WEB. Si usted no está de acuerdo con estos Términos y Condiciones, no puede tener acceso al mismo ni usar el Sitio Web de ninguna otra manera.  
  
1. Derechos de Propiedad. Entre usted y Time Inc., Time Inc. es dueño único y exclusivo, de todos los derechos, título e intereses en y del Sitio Web, de todo el contenido (incluyendo, por ejemplo, audio, fotografías, ilustraciones, gráficos, otros medios visuales, videos, copias, textos, software, títulos, archivos de Onda de choque, etc.), códigos, datos y materiales del mismo, el aspecto y el ambiente, el diseño y la organización del Sitio Web y la compilación de los contenidos, códigos, datos y los materiales en el Sitio Web, incluyendo pero no limitado a, cualesquiera derechos de autor, derechos de marca, derechos de patente, derechos de base de datos, derechos morales, derechos sui generis y otras propiedades intelectuales y derechos patrimoniales del mismo. Su uso del Sitio Web no le otorga propiedad de ninguno de los contenidos, códigos, datos o materiales a los que pueda acceder en o a través del Sitio Web.  
  
  
  
2. Licencia Limitada. Usted puede acceder y ver el contenido del Sitio Web desde su computadora o desde cualquier otro aparato y, a menos de que se indique de otra manera en estos Términos y Condiciones o en el Sitio Web, sacar copias o impresiones individuales del contenido del Sitio Web para su uso personal, interno únicamente. El uso del Sitio Web y de los servicios que se ofrecen en o a través del Sitio Web, son sólo para su uso personal, no comercial.  
  
  
  
3. Uso Prohibido. Cualquier distribución, publicación o explotación comercial o promocional del Sitio Web, o de cualquiera de los contenidos, códigos, datos o materiales en el Sitio Web, está estrictamente prohibida, a menos de que usted haya recibido el previo permiso expreso por escrito del personal autorizado de Time Inc. o de algún otro poseedor de derechos aplicable. A no ser como está expresamente permitido en el presente contrato, usted no puede descargar, informar, exponer, publicar, copiar, reproducir, distribuir, transmitir, modificar, ejecutar, difundir, transferir, crear trabajos derivados de, vender o de cualquier otra manera explotar cualquiera de los contenidos, códigos, datos o materiales en o disponibles a través del Sitio Web. Usted se obliga además a no alterar, editar, borrar, quitar, o de otra manera cambiar el significado o la apariencia de, o cambiar el propósito de, cualquiera de los contenidos, códigos, datos o materiales en o disponibles a través del Sitio Web, incluyendo, sin limitación, la alteración o retiro de cualquier marca comercial, marca registrada, logo, marca de servicios o cualquier otro contenido de propiedad o notificación de derechos de propiedad. Usted reconoce que no adquiere ningún derecho de propiedad al descargar algún material con derechos de autor de o a través del Sitio Web. Si usted hace otro uso del Sitio Web, o de los contenidos, códigos, datos o materiales que ahí se encuentren o que estén disponibles a través del Sitio Web, a no ser como se ha estipulado anteriormente, usted puede violar las leyes de derechos de autor y otras leyes de los Estados Unidos y de otros países, así como las leyes estatales aplicables, y puede ser sujeto a responsabilidad legal por dicho uso no autorizado.  
  
  
  
4. Marcas Comerciales. Las marcas comerciales, logos, marcas de servicios, marcas registradas (conjuntamente las Maras Comerciales) expuestas en el Sitio Web o en los contenidos disponibles a través del Sitio Web son Marcas Comerciales de Time Inc. registradas y no registradas y otras, y no pueden ser usadas con respecto a productos y/o servicios que no estén relacionados, asociados o patrocinados por sus poseedores de derechos y que puedan causar confusión a los clientes, o de alguna manera que denigre o desacredite a sus poseedores de derechos. Todas las Marcas Comerciales que no sean de Time Inc. que aparezcan en el sitio Web o en o a través de los servicios del Sitio Web, si las hubiera, son propiedad de sus respectivos dueños. Nada que esté contenido en el Sitio Web deberá ser interpretado como otorgando, por implicación, desestimación, o de otra manera, alguna licencia o derecho para usar alguna Marca Comercial expuesta en el Sitio Web sin el permiso escrito de Time Inc. o de terceros que puedan ser dueños de dicha Marca Comercial. El mal uso de las Marcas Comerciales expuestas en el Sitio Web o en o a través de cualquiera de los servicios del Sitio Web está estrictamente prohibido.  
  
  
  
5. Información del Usuario. En el curso del uso que usted haga del Sitio Web y/o de los servicios puestos a su disposición en o a través del Sitio Web, se le puede pedir que nos proporcione cierta información personalizada (dicha información en lo sucesivo Información del Usuario). Las políticas de uso y recopilación de información de Time Inc. con respecto a la privacidad de dicha Información del Usuario se establecen en la Política de Privacidad del Sitio Web, la cual está incorporada al mismo como referencia para todos los propósitos. Usted reconoce y acepta ser el único responsable de la exactitud del contenido de la Información del Usuario.  
  
  
  
6. Materiales Presentados. A menos que se solicite específicamente, no pedimos ni deseamos recibir ninguna información confidencial, secreta o patrimonial, ni otro material de usted a través del Sitio Web, por correo electrónico o de cualquier otra manera. Cualquier información, trabajos creativos, demostración, ideas, sugerencias, conceptos, métodos, sistemas, diseños, planes, técnicas u otros materiales que nos haya mandado o presentado (incluyendo, por ejemplo y sin limitación, aquello que usted presenta o envía a nuestros grupos de chateo, tablas de mensajes y/o a nuestros ‘blogs’, o que nos manda vía correo electrónico) (Materiales Presentados) se considerará como no confidencial o secreto y que puede ser usado por nosotros de cualquier manera consistente con la Política de Privacidad del Sitio Web. Al presentarnos o mandarnos Materiales Presentados, usted: (i) representa y garantiza que los Materiales Presentados son originales suyos, que ninguna otra persona tiene ningún derecho sobre ellos, y que cualquier derecho moral sobre los Materiales Presentados ha sido renunciado, y (ii) usted nos concede, a nosotros y a nuestros afiliados, derecho y licencia libres de regalías, sin restricciones, mundiales, perpetuos, irrevocables, no exclusivos y totalmente transferibles, que pueden ser cedidos y sub-licenciados, para usar, copiar, reproducir, modificar, adaptar, publicar, traducir, crear trabajos derivados de, distribuir, ejecutar, exponer e incorporar en otros trabajos cualquiera de los Materiales Presentados (todos o en parte) en cualquier forma, medio o tecnología no conocida o por desarrollar, incluyendo propósitos promocionales y/o comerciales. No podemos ser responsables de conservar ningún Material Presentado proporcionado por usted y podemos borrar o destruir dicho Material Presentado en cualquier momento.  
  
  
  
7. Conducta Prohibida del Usuario. Usted garantiza y está de acuerdo en que, mientras use el Sitio Web y los diversos servicios y artículos que se ofrecen en o a través del Sitio Web, usted no: (a) personalizará a ninguna persona o entidad ni desvirtuará su afiliación con alguna otra persona o entidad; (b) insertará su propio anuncio, posicionamiento de marca o algún otro contenido promocional o el de un tercero en cualquiera de los contenidos, materiales o servicios o materiales del Sitio Web (por ejemplo, sin limitación, en una actualización RSS o en un programa de radio grabado (podcast) recibido de Time Inc. o de algún otro modo a través del Sitio Web), ni usará, redistribuirá, republicará o explotará dichos contenidos o servicios con cualquier otro propósito adicional comercial o promocional; ni (c) intentará ganar acceso no autorizado a otros sistemas de cómputo a través del Sitio Web. Usted no: (i) participará en navegar por la red, en raspar (scraping) la pantalla, raspar (scraping) la base da datos, en recolectar direcciones de correo electrónico, direcciones inalámbricas u otra información personal o de contactos, o cualquier otro medio automático de obtener listas de usuarios u otra información de o a través del sitio Web o de los servicios ofrecidos en o a través del Sitio Web, incluyendo, sin limitación, cualquier información que se encuentre en algún servidor o base de datos relacionada con el Sitio Web o los servicios ofrecidos en o a través del Sitio Web; (ii) obtendrá o intentará obtener acceso no autorizado a los sistemas de cómputo, materiales o información por cualquier medio; (iii) usará el Sitio Web o los servicios puestos a su disposición en o a través del Sitio Web de alguna manera con la intención de interrumpir, dañar, deshabilitar, sobrecargar o deteriorar el Sitio Web o dichos servicios, incluyendo, sin limitación, mandar mensajes masivos no solicitados o inundar servidores con solicitudes; (iv) usará el Sitio Web o los servicios o artículos del Sitio Web en violación de la propiedad intelectual o de otros derechos legales o patrimoniales de Time Inc o de algún tercero; ni (v) usará el Sitio Web o los servicios del Sitio Web en violación de cualquier ley aplicable. Usted se obliga además, a no intentar (o alentar o apoyar el intento de otro) a embaucar, destruir, decodificar, o de otro modo alterar o interferir con el Sitio Web o con los servicios del Sitio Web, o con cualquier contenido del mismo, o hacer cualquier uso no autorizado del mismo. Usted se obliga a no usar el Sitio Web de alguna manera que pudiera dañar, deshabilitar, sobrecargar o deteriorar el Sitio Web o interferir con que cualquier otra persona pueda usar o disfrutar del Sitio Web o de cualquiera de sus servicios. Usted no obtendrá ni intentará obtener algún material o información a través de cualquier medio que no haya sido estipulado o puesto a la disposición del público intencionalmente a través del Sitio Web.  
  
  
  
 8. Foros Públicos. Time Inc. puede, de vez en cuando, tener servicios de mensajería, servicios de chateo, tableros de noticias, blogs, otros foros y otros servicios disponibles en o a través del Sitio Web. Además de cualquier otra normatividad y regulación que podamos publicar con respecto a un servicio en particular, usted se obliga a no cargar, informar, transmitir, distribuir o de otra manera publicar a través del Sitio Web o de cualquier servicio o artículo puesto a la disposición en o a través del Sitio Web, cualquier material que (i) restrinja o inhiba a cualquier otro usuario de usar y disfrutar del Sitio Web o de los servicios del Sitio Web, (ii) sea fraudulento, ilegal, amenazante, abusivo, hostigante, calumnioso, difamatorio, obsceno, vulgar, ofensivo, pornográfico, profano, sexualmente explícito o indecente, (iii) constituya o aliente conductas que pudieran constituir una ofensa criminal, dar lugar a responsabilidad civil o de otro modo violar cualquier ley local, estatal, nacional o internacional, (iv) viole, plagie o infrinja los derechos de terceros incluyendo, sin limitación, derechos de autor, marcas comerciales, secretos comerciales, confidencialidad, contratos, patentes, derechos de privacidad o publicidad o cualquier otro derecho de propiedad, (v) contenga un virus, elemento de espionaje u otro componente dañino, (vi) contenga enlaces fijos, publicidad, cartas de cadenas o esquemas de pirámides de cualquier tipo, o (vii) constituya o contenga indicaciones de origen, endosos o declaraciones de hechos falsos o engañosos. Usted además se obliga a no personificar a cualquier otra persona o entidad, ya sea real o ficticia, incluyendo cualquier persona de Time Inc. Usted tampoco puede ofrecer comprar o vender algún producto o servicio en o a través de sus comentarios presentados en nuestros foros. Solamente usted es responsable del contenido y de las consecuencias de cualquiera de sus actividades.  
  
  
  
 9. Derecho de Monitoreo y Control Editorial. Time Inc. se reserva el derecho, pero no tiene la obligación, de monitorear y/o revisar todos los materiales enviados al Sitio Web o a través de los servicios o artículos del Sitio Web por los usuarios, y Time Inc. no es responsable de dichos materiales enviados por los usuarios. Sin embargo, Time Inc. se reserva el derecho en todo momento de divulgar cualquier información que sea necesaria para satisfacer cualquier ley, reglamento o solicitud gubernamental, o de editar, rehusarse a colocar o a quitar cualquier información o materiales, todos o en parte, que a discreción únicamente de Time Inc. sean censurables o en violación de estos Términos de Uso, de las políticas de Time Inc o de la ley aplicable. También podemos imponer límites sobre ciertos artículos de los foros o restringir su acceso a parte o a todos los foros sin notificación o sanción, si creemos que usted está en incumplimiento de las directrices establecidas en este párrafo, nuestros términos y condiciones o la ley aplicable, o por cualquier otra razón sin notificación o responsabilidad.  
  
  
  
 10. Información Privada o Delicada en Foros Públicos. Es importante recordar que los comentarios presentados en un foro pueden ser registrados y almacenados en múltiples lugares, tanto en nuestro Sitio Web como en otra parte en Internet, los cuales pueden ser accesibles durante mucho tiempo y no se tiene control sobre quién los leerá eventualmente. Es por lo tanto importante que tenga usted cuidado y sea selectivo acerca de la información personal que divulgue acerca de usted y de otros, y en especial, no debe divulgar información delicada, patrimonial o confidencial en sus comentarios en nuestros foros públicos.  
  
  
  
 11. Enlaces con el Sitio Web. Usted está de acuerdo en que si incluye un enlace (link) de cualquier otro sitio web al Sitio Web, dicho enlace se abrirá en una nueva ventana navegadora y se enlazará con la versión completa de una página formateada HTML de este Sitio Web. Usted no tiene permitido enlazarse directamente a ninguna imagen almacenada en el Sitio Web o en nuestros servicios, como sería usar un método de enlace en-línea para provocar que la imagen almacenada por nosotros fuera expuesta en otro sitio web. Usted se obliga a no descargar o usar imágenes almacenadas en este Sitio Web en otro sitio web, con cualquier propósito, incluyendo, sin limitación, publicar dichas imágenes en otro sitio web. Usted se obliga a no enlazarse de cualquier otro sitio web a este Sitio Web de tal manera que el Sitio Web, o cualquier página del Sitio Web, esté enmarcado rodeado u ofuscado por los contenidos, materiales o posicionamientos de marca de cualquier tercero. Nos reservamos todos nuestros derechos bajo la ley para insistir en que cualquier enlace al Sitio Web sea descontinuado y a revocar su derecho a enlazarse al Sitio Web de cualquier otro sitio web, en cualquier momento en el que le mandemos notificación por escrito.  
  
  
  
 12. Indemnización. Usted se obliga a defender, indemnizar y a sacar a Time Inc. y a los directores, funcionarios, empleados y agentes de Time Inc. y sus afiliados en paz y a salvo de cualquier demanda, responsabilidad, costos y gastos, de cualquier naturaleza, incluyendo honorarios de abogados, en que incurriera como resultado del uso del Sitio Web, su colocación o transmisión de cualquier mensaje, contenido, información, software u otros materiales a través del Sitio Web, o su incumplimiento o violación de la ley o de estos Términos y Condiciones. Time Inc. se reserva el derecho, a su propio costo, de asumir la defensa y control exclusivos de cualquier asunto de otra manera sujeto a indemnización por parte suya, y en dicho caso, usted se obliga a cooperar con Time Inc. en la defensa de dicha demanda.  
  
  
  
 13. Órdenes de Productos y Servicios. Podemos poner ciertos productos a la disposición de visitantes y registrados del Sitio Web. Si usted ordena cualquier producto, por el presente documento usted representa y garantiza que tiene 18 años de edad o más. Se obliga a pagar la totalidad de los precios de cualquier compra que haga, ya sea con tarjeta de crédito/débito concurrente con su orden en línea o por otro medio de pago aceptable para Time Inc. Usted se obliga a pagar todos los impuestos aplicables. Si el pago no es recibido por nosotros de parte del emisor de su tarjeta de crédito o débito o de sus agentes, usted se obliga a pagar todas las cantidades debidas al momento de la reclamación por nuestra parte. Algunos productos que usted compra y/o descarga en o a través del Sitio Web pueden estar sujetos a términos y condiciones adicionales que le serán presentados al momento de dicha compra o descarga.  
  
  
  
 14. Sitios Web de Terceros. Usted puede enlazar (link) del Sitio Web a sitios web de terceros y terceros pueden enlazarse al Sitio Web (Sitios Enlazados). Usted reconoce y está de acuerdo en que nosotros no tenemos responsabilidad sobre la información, contenido, productos, servicios, anuncios, códigos u otros materiales que puedan o no puedan ser proporcionados por o a través de los Sitios Enlazados, aún si son propiedad de o son dirigidos por afiliados nuestros. Los enlaces (links) a Sitios Enlazados no constituyen un aval o patrocinio nuestro de dichos sitios web o de la información, contenido, productos, servicios, anuncios, códigos u otros materiales presentados en o a través de dichos sitios web. La inclusión de cualquier enlace a dichos sitios en nuestro Sitio no implica el aval, patrocinio o recomendación de ese sitio de Time Inc. Time Inc. rechaza cualquier responsabilidad por los enlaces (1) de otro sitio web a este Sitio Web y (2) a otro sitio web de este Sitio Web. Time Inc. no puede garantizar los estándares de cualquier sitio web al cual se le proporcionen enlaces en este Sitio Web, ni será Time Inc. responsable de los contenidos de dichos sitios, o de cualquier enlace subsecuente. Time Inc. no representa o garantiza que los contenidos del sitio web de algún tercero sean exactos, que cumplan con la ley estatal o federal, o que cumplan con las leyes de derechos de autor o con otras leyes de propiedad intelectual. Time Inc. tampoco es responsable de cualquier forma de transmisión recibida de cualquier sitio web enlazado. Cualquier confianza depositada en los contenidos de un sitio web de terceros es hecha por su propio riesgo y usted asume todas las responsabilidades y consecuencias que resulten de dicha confianza.  
  
  
  
 15. Agente de Derechos de Autor.  
  
  
  
 Respetamos los derechos de propiedad intelectual de otros y requerimos que las personas que usen el Sitio Web, o los servicios o artículos puestos a su disposición en o a través del Sitio Web, hagan lo mismo. Si usted cree que su trabajo ha sido copiado de un modo que constituya una infracción a los derechos de autor, por favor mande la siguiente información a Agente de Derechos de Autor de Time Inc., designado como tal de conformidad con la Digital Millennium Copyright Act, 17 U.S.C. § 512 (c) (2), nombrado abajo:  
  
  
  
 Su dirección, número de teléfono y dirección de correo electrónico;  
 Una descripción del trabajo con derechos de autor que usted reclama ha sido infringido;  
 Una descripción de donde se localiza el supuesto material infringido;  
 Una declaración suya de que usted cree de buena fe que el uso en disputa no está autorizado por el dueño de los derechos de autor, su agente o la ley;  
 Una firma electrónica o física de la persona autorizada para actuar en nombre del dueño del interés de los derechos de autor; y  
 Una declaración suya, hecha bajo pena de perjurio, de que la información antes mencionada en su notificación es exacta y de que usted es el dueño de los derechos de autor o está autorizado para actuar en nombre del dueño de los derechos de autor.  
  
  
 Agente de Derechos de Autor:  
  
 Tom Schopper  
  
 Time Inc.  
  
 Voz: 212-522-5865  
  
 Fax: 212-467-3320  
  
 e-mail: copyrightdmca@timeinc.com  
  
  
  
 Para Servicio al Cliente no se dirija al Agente de Derechos de Autor, por favor diríjase a:  
  
 Time Customer Service  
  
 3000 University Center Drive  
  
 Tampa, Florida 33612-6408  
  
 Tel: 1-800-541-1000  
  
  
  
 16. RENUNCIA DE GARANTÍAS. EL SITIO WEB, INCLUYENDO, SIN LIMITACIÓN, TODOS LOS SERVICIOS, CONTENIDOS, FUNCIONES Y MATERIALES PROPORCIONADOS A TRAVÉS DEL SITIO WEB, SON PROPORCIONADOS TAL COMO ESTÉN, COMO SE PONEN A LA DISPOSICIÓN, SIN GARANTÍA DE NINGÚN TIPO, YA SEA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITACIÓN, CUALQUIER GARANTÍA DE INFORMACIÓN, DATOS, SERVICIOS DE PROCESAMIENTO DE DATOS, ACCESO ILIMITADO O ININTERRUMPIDO, CUALQUIER GARANTÍA CON RESPECTO A LA DISPONIBILIDAD, INTERPRETABILIDAD, NO INTERPRETABILIDAD, EXACTITUD, PRECISIÓN, RECTITUD, PERFECCIÓN, CONCLUSIÓN, UTILIDAD, O CONTENIDO DE LA INFORMACIÓN, Y CUALQUIER GARANTÍA DE TÍTULO, NO-INFRACCIÓN, COMERCIABILIDAD O APTITUD PARA UN PROPÓSITO EN PARTICULAR, Y EN EL PRESENTE RENUNCIAMOS A CUALQUIERA Y A TODAS DICHAS GARANTÍAS, EXPRESAS E IMPLÍCITAS. NO GARANTIZAMOS QUE EL SITIO WEB O LOS SERVICIOS, CONTENIDOS, FUNCIONES O MATERIALES PROPORCIONADOS A TRAVÉS DEL SITIO WEB SERÉN OPORTUNOS, SEGUROS, ININTERRUMPIDOS O LIBRES DE ERROR, O QUE LOS DEFECTOS SERÉN CORREGIDOS. NO DAMOS GARANTÍA DE QUE EL SITIO WEB O DE QUE LOS SERVICIOS PROPORCIONADOS CUMPLIRÉN CON LOS REQUERIMIENTOS DE LOS USUARIOS. NINGÚN CONSEJO, RESULTADOS O INFORMACIÓN, YA SEA ORAL O ESCRITO, QUE USTED OBTENGA DE NOSOTROS A TRAVÉS DEL SITIO WEB CREARÉ ALGUNA GARANTÍA QUE NO HAYA SIDO ESTABLECIDA EXPRESAMENTE EN EL PRESENTE. TIME INC. TAMPOCO ASUME NINGUNA RESPONSABILIDAD, Y NO SERÉ RESPONSABLE DE CUALQUIER DAÑO A, O DE CUALQUIER VIRUS QUE PUDIERA INFECTAR SU EQUIPO A CAUSA DE SU ACCESO A, USO DE, O NAVEGACIÓN EN EL SITIO WEB O POR DESCARGAR CUALQUIERA DE LOS MATERIALES, DATOS, TEXTOS, IMÉGENES, CONTENIDOS DE VIDEO O CONTENIDOS DE AUDIO DEL SITIO WEB. SI USTED NO ESTÉ SATISFECHO CON EL SITIO WEB, SU ÚNICA SOLUCIÓN ES DESCONTINUAR USANDO EL SITIO WEB.  
  
  
  
 TRATAMOS DE ASEGURAR QUE LA INFORMACIÓN PUBLICADA EN EL SITIO WEB ES CORRECTA Y ACTUALIZADA. NOS RESERVAMOS EL DERECHO DE CAMBIAR O HACER CORRECCIONES A CUALQUIER INFORMACIÓN PROPORCIONADA EN EL SITIO WEB EN CUALQUIER MOMENTO Y SIN NINGÚN AVISO PREVIO. TIME INC. NI AVALA NI ES RESPONSABLE DE LA EXACTITUD O VERACIDAD DE CUALQUIER OPINIÓN, CONSEJO O DECLARACIÓN EN EL SITIO WEB, NI DE CUALQUIER PUBLICACIÓN OFENSIVA, DIFAMATORIA, OBSCENA, INDECENTE, ILEGAL O VIOLATORIA HECHA EN EL MISMO POR CUALQUIER PERSONA A NO SER UN EMPLEADO PORTAVOZ AUTORIZADO DE TIME INC. EN SU CARÉCTER OFICIAL (INCLUYENDO, SIN LIMITACIÓN, OTROS USUARIOS DEL SITIO WEB). ES SU RESPONSABILIDAD EVALUAR LA EXACTITUD, CONCLUSIÓN O UTILIDAD DE CUALQUIER INFORMACIÓN, OPINIÓN, CONSEJO U OTRO CONTENIDO DISPONIBLE A TRAVÉS DEL SITIO WEB. POR FAVOR BUSQUE EL CONSEJO DE PROFESIONALES, SEGÚN SEA APROPIADO, CON RESPECTO A LA EVALUACIÓN DE CUALQUIER INFORMACIÓN, OPINIÓN, CONSEJO U OTRO CONTENIDO ESPECÍFICO, INCLUYENDO PERO NO LIMITADO A, INFORMACIÓN, OPINIÓN, CONSEJO U OTRO CONTENIDO FINANCIERO, DE SALUD O DE ESTILO DE VIDA.  
  
  
  
 PREVIO A LA REALIZACIÓN DE UNA COMPRA O VENTA DE CUALQUIER VALOR O INVERSIÓN, LE ACONSEJAMOS QUE CONSULTE CON SU CORREDOR O CON OTRO ASESOR FINANCIERO PARA VERIFICAR PRECIOS Y OTRA INFORMACIÓN. NO TENDREMOS NINGUNA RESPONSABILIDAD DE DECISIONES DE INVERSIÓN BASADAS EN, O DE RESULTADOS OBTENIDOS DE, EL CONTENIDO PROPORCIONADO EN EL PRESENTE. NADA CONTENIDO EN EL SITIO WEB SERÉ INTERPRETADO COMO CONSEJO DE INVERSIÓN. TIME INC. NO ES UN CORREDOR DE BOLSA REGISTRADO NI UN CONSULTOR DE INVERSIONES Y NO DA CONSEJOS SOBRE INVERSIONES NI RECOMIENDA UN PRODUCTO SOBRE OTRO.  
  
  
  
 SIN LIMITACIÓN DE LO ANTES MENCIONADO EN ESTA SECCIÓN, TIME INC, Y SUS AFILIADOS, PROVEEDORES Y PERSONAS QUE OTORGAN LAS LICENCIAS NO DAN GARANTÍAS NI REPRESENTACIONES CON RESPECTO A CUALQUIER PRODUCTO O SERVICIO ORDENADO O PROPORCIONADO VÍA EL SITIO WEB, Y EN EL PRESENTE RENUNCIAN, Y USTED EN EL PRESENTE DESISTE, DE CUALQUIERA Y DE TODAS LAS GARANTÍAS Y REPRESENTACIONES HECHAS A LITERATURA EN PRODUCTOS O SERVICIOS, A PREGUNTAS HECHAS FRECUENTEMENTE EN DOCUMENTOS Y DE ALGUNA OTRA MANERA EN EL SITIO WEB O EN CORRESPONDENCIA CON TIME INC. O CON SUS AGENTES. CUALQUIER PRODUCTO O SERVICIO ORDENADO O PROPORCIONADO VÍA EL SITIO WEB ES PROPORCIONADO POR TIME INC. TAL COMO ESTÉ, EXCEPTO AL GRADO, SI SE DIERA EL CASO, DE ESTAR DE OTRO MODO ESTABLECIDO POR ESCRITO, EN UN CONTRATO DE LICENCIA O VENTA CELEBRADO POR SEPARADO ENTRE USTED Y TIME INC. O SU PROVEEDOR O LA PERSONA QUE OTORGA LA LICENCIA.  
  
 17. LÍMITE DE RESPONSABILIDAD. EN NINGÚN CASO, INCLUYENDO PERO NO LIMITADO A NEGLIGENCIA, SERÉ RESPONSABLE TIME INC., O CUALQUIERA DE SUS DIRECTORES, FUNCIONARIOS, EMPLEADOS, AGENTES O PROVEEDORES, DEL CONTENIDO O DE LOS SERVICIOS (CONJUNTAMENTE, LAS ENTIDADES PROTEGIDAS) DE CUALQUIER DAÑO DIRECTO, INDIRECTO, ESPECIAL, INCIDENTAL, CONSECUENTE, EJEMPLAR O PUNITIVO COMO RESULTADO DE, O DIRECTA O INDIRECTAMENTE RELACIONADO CON, EL USO DE, O LA INCAPACIDAD DE USAR EL SITIO WEB O LOS CONTENIDOS, MATERIALES Y FUNCIONES RELACIONADOS CON EL MISMO, SU PROVISIÓN DE INFORMACIÓN VÍA EL SITIO WEB, NEGOCIOS PERDIDOS O VENTAS PERDIDAS, AÚN CUANDO DICHA ENTIDAD PROTEGIDA HAYA SIDO ASESORADA SOBRE LA POSIBILIDAD DE DICHOS DAÑOS. ALGUNAS JURISDICCIONES NO PERMITEN LA LIMITACIÓN O EXCLUSIÓN DE RESPONSABILIDAD POR DAÑOS INCIDENTALES O INDIRECTOS ASÍ QUE ALGUNOS DE LAS LIMITACIONES ARRIBA MENCIONADAS NO APLICAN A ALGUNOS USUARIOS. EN NINGÚN CASO SERÉN RESPONSABLES LAS ENTIDADES PROTEGIDAS POR O EN RELACIÓN CON CUALQUIER CONTENIDO PUBLICADO, TRANSMITIDO, INTERCAMBIADO O RECIBIDO POR O EN NOMBRE DE CUALQUIER USUARIO O DE OTRA PERSONA EN O A TRAVÉS DEL SITIO WEB. EN NINGÚN CASO LA SUMA TOTAL DE LAS RESPONSABILIDADES DE LAS ENTIDADES PROTEGIDAS HACIA USTED, POR TODOS LOS DAÑOS, PÉRDIDAS Y CAUSAS DE ACCIÓN PENAL (YA SEA POR CONTRATO O POR AGRAVIO, INCLUYENDO PERO NO LIMITADO A, NEGLIGENCIA O DE ALGUNA OTRA MANERA) QUE RESULTEN DE LOS TÉRMINOS Y CONDICIONES O DEL USO QUE USTED HAGA DEL SITIO WEB, DEBERÉ EXCEDER EN LA SUMA TOTAL, LA CANTIDAD, SI LA HUBIERA, PAGADA POR USTED A TIME INC. POR EL USO DEL SITIO WEB O POR LA COMPRA DE PRODUCTOS VÍA EL SITIO WEB.  
  
  
  
 18. Ataques Fotosensibles. Un muy pequeño porcentaje de personas pueden experimentar un ataque al ser expuestas a ciertas imágenes visuales, como luces o dibujos intermitentes que pueden aparecer en juegos de video o en otros contenidos electrónicos o en línea. Aún personas que no tienen historia de ataques o de epilepsia pueden tener una condición no diagnosticada que puede causar estos ataques epilépticos fotosensibles al estar viendo juegos de video u otros contenidos electrónicos. Estos ataques tienen una variedad de síntomas, incluyendo mareos, desorientación, confusión, pérdida momentánea de conciencia, crispación de ojos o cara, visión alterada o tirones o sacudidas de brazos o piernas. Si usted experimenta cualquiera de los síntomas anteriormente mencionados, o si usted o su familia tiene historia de ataques o epilepsia, debe de dejar inmediatamente de usar el Sitio Web y consultar un doctor.  
  
  
  
 19. Leyes Aplicables. Nosotros controlamos y operamos el Sitio Web desde nuestras oficinas en los Estados Unidos de América. No representamos a los materiales en el Sitio Web como apropiados o disponibles para su uso en otros lugares. Las personas que escojan acceder al sitio Web desde otros lugares lo harán por su propia iniciativa y son responsables del cumplimiento de las leyes locales, si y al grado en que las leyes locales sean aplicables. Todas las partes sujetas a estos términos y condiciones renuncian a sus derechos respectivos a un juicio con jurado.  
  
  
  
 20. Término. Time Inc. puede terminar, cambiar, suspender o descontinuar cualquier aspecto del Sitio Web o de los servicios del Sitio Web en cualquier momento. Time Inc. puede restringir, suspender o terminar su acceso al Sitio Web y/o a sus servicios si creemos que usted está en incumplimiento de nuestros términos y condiciones o de la ley aplicable, o por cualquier otra razón sin notificación o responsabilidad. Time Inc. mantiene una política que estipula la terminación, en circunstancias apropiadas, de los privilegios de uso del Sitio Web para usuarios que son violadores repetitivos de los derechos de propiedad intelectual.  
  
  
  
 21. Cambios de Términos de Uso. Time Inc. se reserva el derecho, a su sola discreción, de cambiar, modificar, añadir o quitar cualquier porción de los Términos y Condiciones, toda o en parte, en cualquier momento. Los cambios en los Términos y Condiciones serán efectivos cuando se publiquen. La continuación del uso del Sitio Web y/o de los servicios puestos a disposición en o a través del Sitio Web después de haber sido publicado cualquier cambio, será considerado como aceptación de esos cambios.  
 22. Misceláneos. Los Términos y Condiciones y la relación entre usted y nosotros, serán regidos por las leyes del Estado de Nueva York, Estados Unidos de América, sin consideración a conflictos de disposiciones de ley. Usted se obliga a que cualquier causa de acción legal que resulte bajo los Términos y Condiciones será iniciada y oída en la corte apropiada en el Estado de Nueva York, Condado de Nueva York, Estados Unidos de América. Usted se obliga a someterse a la jurisdicción personal y exclusiva de las cortes localizadas dentro del Condado de Nueva York en el Estado de Nueva York. Nuestra falla al ejercer y hacer valer cualquier derecho o disposición de los Términos y Condiciones no constituirá una renuncia a dicho derecho o disposición. Si cualquier disposición de los Términos y Condiciones es encontrada inválida por una corte de jurisdicción competente, las partes, sin embargo, están de acuerdo en que la corte deberá esforzarse en aplicar las intenciones de las partes como están reflejadas en la disposición, y las otras disposiciones de los Términos y Condiciones permanecen vigentes.  
 23. Términos Suplementarios.  
 Notificación de Getty Images:  
 Las imágenes inmóviles y representaciones visuales de Getty Images (incluyendo, sin limitación, Sport, MLB, NHLI, AFP Sport y Bongarts Sports de propiedad absoluta de Getty Images) no pueden ser republicadas, retransmitidas, reproducidas, descargadas o usadas de cualquier otra manera, excepto para descarga para uso personal, no comercial.  
 Estos Términos y Condiciones fueron corregidos 02/04/2009."  
 tools:layout\_editor\_absoluteX="105dp"  
 tools:layout\_editor\_absoluteY="118dp" />  
  
 <Button  
 android:id="@+id/btnAtrasTerminos"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:background="@drawable/redondeadobutton"  
 android:text="Atras"  
 android:textColor="#fff"/>  
 </LinearLayout>  
 </ScrollView>  
</LinearLayout>

DI45. Content\_animal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.animal"  
 android:background="#d6d6d6">  
  
 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.animal"  
 tools:showIn="@layout/app\_bar\_animal"  
 android:orientation="vertical"  
 android:layout\_marginBottom="55dp">  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/imgAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:src="@mipmap/ic\_launcher"  
 android:layout\_marginTop="50dp"/>  
  
 <TextView  
 android:id="@+id/txtNombreAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:text="Ruben Palomo"  
 android:textSize="50dp"  
 android:textStyle="bold"  
 android:textColor="#151414"  
 android:background="#c94db6ac"/>  
 </RelativeLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <LinearLayout  
 android:id="@+id/ly1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:orientation="vertical"  
 android:background="@drawable/datosanimalfondo">  
 <TextView  
 android:id="@+id/txtGenero"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="..."  
 android:gravity="center"  
 android:textSize="25dp"  
 android:textColor="#000"/>  
 <ImageView  
 android:id="@+id/imgGenero"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:src="@drawable/gender"  
 android:padding="10dp"/>  
 </LinearLayout>  
 <LinearLayout  
 android:id="@+id/ly2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:orientation="vertical"  
 android:background="@drawable/datosanimalfondo">  
 <TextView  
 android:id="@+id/txtEdadAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="..."  
 android:gravity="center"  
 android:textSize="25dp"  
 android:textColor="#000"/>  
 <ImageView  
 android:id="@+id/imgAnos"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:src="@drawable/balloon"  
 android:padding="10dp"/>  
 </LinearLayout>  
 <LinearLayout  
 android:id="@+id/ly3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:orientation="vertical"  
 android:background="@drawable/datosanimalfondo">  
 <TextView  
 android:id="@+id/txtVacuna"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="..."  
 android:gravity="center"  
 android:textSize="25dp"  
 android:textColor="#000"/>  
 <ImageView  
 android:id="@+id/imgVacuna"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:padding="10dp"  
 android:src="@drawable/vaccine"/>  
 </LinearLayout>  
 <LinearLayout  
 android:id="@+id/ly4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:orientation="vertical"  
 android:background="@drawable/datosanimalfondo">  
 <TextView  
 android:id="@+id/txtChip"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="..."  
 android:gravity="center"  
 android:textSize="25dp"  
 android:textColor="#000"/>  
 <ImageView  
 android:id="@+id/imgChip"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:src="@drawable/chip"  
 android:padding="10dp"/>  
 </LinearLayout>  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:background="@drawable/redondeadoenviarcorreo"  
 android:gravity="center"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:padding="15dp">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <TextView  
 android:id="@+id/txtInformacion"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Informacion: "  
 android:textSize="30dp"  
 android:textColor="#151414"/>  
  
 <TextView  
 android:id="@+id/txtEmailAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=" hola@gmail.com"  
 android:textSize="20dp"  
 android:textColor="#212020"/>  
  
 <TextView  
 android:id="@+id/txtNumeroAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=" 6666666666"  
 android:textSize="20dp"  
 android:textColor="#212020"/>  
 </LinearLayout>  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:background="@drawable/redondeadoenviarcorreo"  
 android:gravity="center"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:padding="15dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <TextView  
 android:id="@+id/txtInfoAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Datos: "  
 android:textSize="30dp"  
 android:textColor="#151414"/>  
 <TextView  
 android:id="@+id/txtTipoAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Tipo"  
 android:textSize="20dp"  
 android:textColor="#212020"/>  
 <TextView  
 android:id="@+id/txtDescripcionAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Descripcion..."  
 android:textSize="18dp"  
 android:textColor="#151414"/>  
 </LinearLayout>  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@drawable/redondeadoenviarcorreo"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:padding="15dp">  
  
 <TextView  
 android:id="@+id/txtEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Email: "  
 android:textSize="23dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="Tu email..."  
 android:layout\_marginBottom="10dp"/>  
  
 <TextView  
 android:id="@+id/txtMensaje"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Mensaje: "  
 android:textSize="23dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtMensaje"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="Tu mensaje..."  
 android:layout\_marginBottom="10dp"/>  
  
 <Button  
 android:id="@+id/btnEnviarAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Enviar Correo"  
 android:textSize="20dp"  
 android:textColor="#ffffff"  
 android:background="@drawable/redondeadobutton"/>  
 </LinearLayout>  
 </LinearLayout>  
 </ScrollView>  
 </LinearLayout>  
 <android.support.design.widget.BottomNavigationView  
 android:id="@+id/navigation"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="0dp"  
 android:layout\_marginStart="0dp"  
 android:background="#bf44a198"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:menu="@menu/navigationanimal"/>  
</android.support.constraint.ConstraintLayout>

DI46. Content\_ayuda.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.ayuda"  
 tools:showIn="@layout/app\_bar\_ayuda"  
 android:padding="20dp"  
 android:orientation="vertical"  
 android:background="@drawable/fondoayudainfo">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:layout\_marginLeft="36dp"  
 android:layout\_marginRight="33dp">  
  
 <TextView  
 android:id="@+id/enunciadoAyuda"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Ayuda"  
 android:layout\_marginBottom="10dp"  
 android:textSize="50dp"  
 android:gravity="center"  
 android:textColor="#151414"/>  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 <TextView  
 android:id="@+id/txtAyuda"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Esta aplicacion trata de ayudar a los animales mas desfavorecidos a encontrar un nuevo hogar donde vivan muy bien.\n\n  
Con esta aplicacion sin registrarte podras contactar para adoptar animales y ver los refugios de animales que hay.\n\n  
Con esta aplicacion registrandote aparte de lo anterior podras subir animales a dar en adopcion y borrar tus animales.\n\n  
Podras ponerte en contacto con nosotros en el apartado contacto.\n\n  
Gracias por leerlo, todo por ayudar a los animales."  
 android:textSize="27dp"  
 android:textColor="#212020"  
 android:background="@drawable/datoscuenta"  
 android:padding="5dp"/>  
 </ScrollView>  
 </LinearLayout>  
</LinearLayout>

DI47. Content\_contacto.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.contacto"  
 tools:showIn="@layout/app\_bar\_contacto"  
 android:padding="20dp"  
 android:orientation="vertical"  
 android:background="@drawable/fondocontacto">  
  
 <TextView  
 android:id="@+id/enunciadoContacto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Contacto"  
 android:layout\_marginBottom="30dp"  
 android:textSize="50dp"  
 android:gravity="center"  
 android:textColor="#151414"/>  
  
 <TextView  
 android:id="@+id/txtNombreContacto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Nombre:"  
 android:textSize="27dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtNombreContacto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:textSize="20dp"  
 android:background="@drawable/redondeadoedt"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="Tu nombre..."  
 android:layout\_marginBottom="10dp" />  
  
 <TextView  
 android:id="@+id/txtEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Email: "  
 android:textSize="27dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:textSize="20dp"  
 android:background="@drawable/redondeadoedt"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="Tu email..."  
 android:layout\_marginBottom="10dp" />  
  
 <TextView  
 android:id="@+id/txtMensaje"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Mensaje: "  
 android:textSize="27dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtMensaje"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:textSize="20dp"  
 android:background="@drawable/redondeadoedt"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="Tu mensaje..."  
 android:layout\_marginBottom="10dp" />  
  
 <Button  
 android:id="@+id/btnEnviarContacto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Enviar"  
 android:textSize="20dp"  
 android:textColor="#ffffff"  
 android:onClick="enviarCorreo"  
 android:background="@drawable/redondeadobutton"/>  
</LinearLayout>

DI48. Content\_cuenta.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.cuenta"  
 tools:showIn="@layout/app\_bar\_cuenta"  
 android:orientation="vertical"  
 android:background="#d6d6d6">  
  
 <TextView  
 android:id="@+id/txtIdCuenta"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="id:"  
 android:textSize="6dp"  
 android:textStyle="bold"  
 android:textColor="#383838"  
 android:layout\_marginLeft="10dp"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_margin="10dp"  
 android:padding="10dp"  
 android:background="@drawable/datoscuenta">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/imgCuenta"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:background="@drawable/redondeadoimg"  
 android:src="@mipmap/ic\_launcher"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:layout\_marginLeft="10dp">  
 <TextView  
 android:id="@+id/txtNombreCuenta"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:text="Ruben Palomo"  
 android:textSize="35dp"  
 android:textStyle="bold"  
 android:textColor="#151414"/>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="3dp"  
 android:background="#3b8e86"/>  
  
 <TextView  
 android:id="@+id/txtEmailCuenta"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:text="Ruben Palomo"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 android:textColor="#151414"/>  
 </LinearLayout>  
  
 </LinearLayout>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="3dp"  
 android:background="#3b8e86"  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginRight="5dp"/>  
  
 <GridView  
 android:id="@+id/listAnimalesCuenta"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:numColumns="2"  
 android:horizontalSpacing="10dp"  
 android:verticalSpacing="5dp"  
 android:layout\_margin="10dp"  
 android:padding="10dp"  
 android:background="@drawable/datoscuenta"/>  
  
</LinearLayout>

DI49. Content\_info.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.info"  
 tools:showIn="@layout/app\_bar\_info"  
 android:padding="30dp"  
 android:orientation="vertical"  
 android:background="@drawable/fondoayudainfo">  
  
<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:layout\_marginLeft="28dp"  
 android:layout\_marginRight="27dp">  
 <TextView  
 android:id="@+id/enunciadoInfo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Sobre nosotros: "  
 android:layout\_marginBottom="10dp"  
 android:textSize="50dp"  
 android:gravity="center"  
 android:textColor="#151414"/>  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 <TextView  
 android:id="@+id/txtAyuda"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Buenas, me llamo Ruben Palomo Yuste de 19 años de edad, soy un estudiante de desarrollo de aplicaciones multiplataforma.\n\n  
Esta aplicacion es un trabajo de final de grado. La hago para ayudar a los animales mas desfavorecidos que por mala suerte hoy en dia hay muchos.\n\n  
Con esta aplicacion espero ayudar a muchos animales en mal estado.\n\n  
Gracias por descargaros esta aplicacion."  
 android:textSize="27dp"  
 android:textColor="#212020"  
 android:background="@drawable/datoscuenta"  
 android:padding="5dp"/>  
 </ScrollView>  
</LinearLayout>  
</LinearLayout>

DI50. Content\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.login"  
 tools:showIn="@layout/app\_bar\_login"  
 android:padding="50dp"  
 android:orientation="vertical"  
 android:background="@drawable/fondo">  
  
 <TextView  
 android:id="@+id/enunciadoLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Login"  
 android:layout\_marginBottom="10dp"  
 android:textSize="50dp"  
 android:gravity="center"  
 android:textColor="#151414"/>  
  
 <ImageView  
 android:layout\_width="171dp"  
 android:layout\_height="84dp"  
 android:layout\_gravity="center"  
 android:src="@drawable/icono" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:orientation="vertical"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginRight="25dp"  
 android:layout\_marginLeft="25dp">  
  
 <TextView  
 android:id="@+id/txtEmailLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Email: "  
 android:textSize="30dp"  
 android:textColor="#212020"  
 android:gravity="center"/>  
  
 <EditText  
 android:id="@+id/edtEmailLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:inputType="textEmailAddress"  
 android:hint="Tu email..."  
 android:textSize="20dp"  
 android:background="@drawable/redondeadoedt"  
 android:layout\_marginBottom="20dp"/>  
  
 <TextView  
 android:id="@+id/txtPasswordLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Contraseña: "  
 android:textSize="30dp"  
 android:textColor="#212020"  
 android:gravity="center"/>  
  
 <EditText  
 android:id="@+id/edtPasswordLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:textSize="20dp"  
 android:background="@drawable/redondeadoedt"  
 android:ems="10"  
 android:inputType="textPassword"  
 android:hint="Tu contraseña..."  
 android:layout\_marginBottom="30dp"/>  
  
 <Button  
 android:id="@+id/btnEntrarLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="20dp"  
 android:background="@drawable/redondeadobutton"  
 android:textColor="#ffffff"  
 android:text="Entrar"  
 android:onClick="loguearse"/>  
 </LinearLayout>  
  
</LinearLayout>

DI51. Content\_pagina\_princial.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.paginaPrincipal"  
 tools:showIn="@layout/app\_bar\_pagina\_principal"  
 android:orientation="vertical"  
 android:background="#d6d6d6">  
  
 <ImageButton  
 android:id="@+id/btnBuscar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="53dp"  
 android:background="@drawable/buscador"  
 android:text="Busca"  
 android:onClick="VerFiltros"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/datoscuenta"  
 android:layout\_margin="10dp">  
 <GridView  
 android:id="@+id/listAnimalesPrincipal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="20dp"  
 android:numColumns="2"  
 android:horizontalSpacing="10dp"  
 android:verticalSpacing="5dp"/>  
 </LinearLayout>  
  
</LinearLayout>

DI52. Content\_refugio.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.refugio"  
 tools:showIn="@layout/app\_bar\_refugio"  
 android:orientation="vertical"  
 android:background="#d6d6d6">  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:layout\_marginBottom="55dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_margin="10dp"  
 android:padding="10dp"  
 android:background="@drawable/datoscuenta">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/imgRefugio"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:src="@mipmap/ic\_launcher"  
 android:layout\_marginRight="10dp"  
 android:background="@drawable/redondeadoimg"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:layout\_marginLeft="10dp">  
 <TextView  
 android:id="@+id/txtNombreRefugio"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:text="Ruben Palomo"  
 android:textSize="35dp"  
 android:textStyle="bold"  
 android:textColor="#151414"/>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="3dp"  
 android:background="#3b8e86"/>  
  
 <TextView  
 android:id="@+id/txtEmailRefugio"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:text="Ruben Palomo"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 android:textColor="#151414"/>  
 </LinearLayout>  
  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:background="@drawable/redondeadoenviarcorreo"  
 android:gravity="center"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:padding="15dp">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <TextView  
 android:id="@+id/txtInformacion"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Informacion: "  
 android:textSize="30dp"  
 android:textColor="#151414"/>  
 <TextView  
 android:id="@+id/txtNumeroRefugio"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=" 6666666666"  
 android:textSize="20dp"  
 android:textColor="#212020"/>  
 <TextView  
 android:id="@+id/txtDireccionRefugio"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=" 6666666666"  
 android:textSize="20dp"  
 android:textColor="#212020"/>  
 </LinearLayout>  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@drawable/redondeadoenviarcorreo"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:padding="15dp">  
  
 <TextView  
 android:id="@+id/txtEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Email: "  
 android:textSize="23dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="Tu email..."  
 android:layout\_marginBottom="10dp"/>  
  
 <TextView  
 android:id="@+id/txtMensaje"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Mensaje: "  
 android:textSize="23dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtMensaje"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="Tu mensaje..."  
 android:layout\_marginBottom="10dp"/>  
  
 <Button  
 android:id="@+id/btnEnviarAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Enviar Correo"  
 android:textSize="20dp"  
 android:textColor="#ffffff"  
 android:background="@drawable/redondeadobutton"/>  
 </LinearLayout>  
 </LinearLayout>  
 </ScrollView>  
 <android.support.design.widget.BottomNavigationView  
 android:id="@+id/navigation"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="0dp"  
 android:layout\_marginStart="0dp"  
 android:background="#bf44a198"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:menu="@menu/navigationanimal"/>  
  
</android.support.constraint.ConstraintLayout>

DI53. Content\_ refugios.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.refugios"  
 tools:showIn="@layout/app\_bar\_refugios"  
 android:background="#d6d6d6">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/datoscuenta"  
 android:layout\_margin="10dp"  
 android:padding="10dp">  
 <GridView  
 android:id="@+id/listRefugios"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:numColumns="3"  
 android:horizontalSpacing="10dp"  
 android:verticalSpacing="5dp"/>  
 </LinearLayout>  
  
</LinearLayout>

DI54. Content\_registro.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.registro"  
 tools:showIn="@layout/app\_bar\_registro"  
 android:padding="40dp"  
 android:orientation="vertical"  
 android:background="@drawable/fondoregisto">  
  
 <TextView  
 android:id="@+id/enunciadoRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Registro"  
 android:layout\_marginBottom="20dp"  
 android:textSize="50dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="#151414"/>  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@drawable/datoscuenta"  
 android:padding="10dp"  
 >  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/imgSubirAnimal"  
 android:src="@drawable/icono"  
 android:layout\_width="90dp"  
 android:layout\_height="90dp" />  
 <Button  
 android:id="@+id/btnAvatar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:layout\_gravity="center"  
 android:text="Selecciona tu avatar"  
 android:background="@drawable/redondeadobutton"  
 android:textSize="20dp"  
 android:textColor="#ffffff"/>  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/txtNombreRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Nombre Completo: "  
 android:textSize="26dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtNombreRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Tu nombre..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPersonName" />  
  
 <TextView  
 android:id="@+id/txtNacimientoRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Fecha de Nacimiento: "  
 android:textSize="26dp"  
 android:textColor="#212020"/>  
  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginBottom="10dp"  
 android:layout\_gravity="center">  
  
 <Spinner  
 android:id="@+id/spDiaRegistro"  
 android:layout\_width="60dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/redondeadoedt"  
 android:layout\_marginRight="10dp" />  
  
 <Spinner  
 android:id="@+id/spMesRegistro"  
 android:layout\_width="110dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/redondeadoedt"  
 android:layout\_marginRight="10dp"/>  
  
 <Spinner  
 android:id="@+id/spAnoRegistro"  
 android:layout\_width="90dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/redondeadoedt"  
 android:layout\_marginRight="10dp"/>  
  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/txtNombreUserRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Nombre de Usuario: "  
 android:textSize="26dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtNombreUserRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Tu nombre usuario..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPersonName" />  
  
 <TextView  
 android:id="@+id/txtEmailRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Email: "  
 android:textSize="26dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtEmailRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Tu email..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textEmailAddress" />  
  
  
 <TextView  
 android:id="@+id/txtPasswordRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Contraseña: "  
 android:textSize="26dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtPasswordRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Tu contraseña..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPassword" />  
  
 <EditText  
 android:id="@+id/edtPasswordRepiteRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Repite contraseña..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPassword" />  
  
 <TextView  
 android:id="@+id/txtTelefonoRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Telefono: "  
 android:textSize="26dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtTelefonoRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Tu telefono..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="number" />  
  
 <TextView  
 android:id="@+id/txtDireccionRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Direccion: "  
 android:textSize="26dp"  
 android:textColor="#212020"/>  
  
 <EditText  
 android:id="@+id/edtDireccionRegistro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Tu direccion..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="text" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginBottom="10dp">  
 <CheckBox  
 android:id="@+id/chkAceptarTerminosRegistro"  
 android:layout\_width="180dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="Soy mayor de edad y acepto los terminos"  
 android:textSize="18dp"  
 android:textColor="#212020"/>  
  
 <Button  
 android:id="@+id/btnVerTerminos"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:text="Ver Terminos"  
 android:textSize="15dp"  
 android:layout\_gravity="right"  
 android:gravity="center"  
 android:background="@drawable/redondeadobutton"  
 android:textColor="#ffffff"  
 android:onClick="VerTerminos"/>  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/btnRegistrar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:text="Registrarse"  
 android:textSize="20dp"  
 android:background="@drawable/redondeadobutton"  
 android:textColor="#ffffff"  
 android:onClick="registrarse"/>  
 </LinearLayout>  
 </ScrollView>  
</LinearLayout>

DI55. Content\_subiranimal.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.subiranimal"  
 tools:showIn="@layout/app\_bar\_subiranimal"  
 android:orientation="vertical"  
 android:padding="10dp"  
 android:background="#d6d6d6">  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@drawable/datoscuenta"  
 android:padding="10dp">  
  
 <TextView  
 android:id="@+id/enunciadoSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Nuevo animal"  
 android:layout\_marginBottom="20dp"  
 android:textSize="50dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="#151414"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/imgSubirAnimal"  
 android:src="@drawable/icono"  
 android:layout\_width="90dp"  
 android:layout\_height="90dp"  
 android:layout\_marginRight="5dp"/>  
 <Button  
 android:id="@+id/btnAvatar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:layout\_gravity="center"  
 android:text="Selecciona avatar del animal"  
 android:background="@drawable/redondeadobutton"  
 android:textSize="20dp"  
 android:textColor="#ffffff"/>  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/txtNombreAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Nombre mascota: "/>  
  
 <EditText  
 android:id="@+id/edtNombreAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Su nombre..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPersonName"/>  
  
 <TextView  
 android:id="@+id/txtTipoAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Tipo de la mascota: "/>  
  
 <Spinner  
 android:id="@+id/spTipoAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPersonName"/>  
  
 <TextView  
 android:id="@+id/txtRazaAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Raza de la mascota: "/>  
  
 <EditText  
 android:id="@+id/edtRazaAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Su raza..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPersonName"/>  
  
 <TextView  
 android:id="@+id/txtDescripcionAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Descripcion de la mascota: "/>  
  
 <EditText  
 android:id="@+id/edtDescripcionAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Datos varios..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="textPersonName"/>  
  
 <TextView  
 android:id="@+id/txtEdadAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Edad de la mascota: "/>  
  
 <EditText  
 android:id="@+id/edtEdadAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="2..."  
 android:background="@drawable/redondeadoedt"  
 android:inputType="number"/>  
  
 <TextView  
 android:id="@+id/txtGeneroAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="Genero de la mascota: "/>  
  
 <RadioGroup  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginBottom="10dp">  
 <RadioButton  
 android:id="@+id/rbGeneroMAnimalSubir"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_marginRight="10dp"  
 android:textSize="18dp"  
 android:text="Macho" />  
 <RadioButton  
 android:id="@+id/rbGeneroHAnimalSubir"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:textSize="18dp"  
 android:text="Hembra" />  
 </RadioGroup>  
  
 <TextView  
 android:id="@+id/txtVacunaAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="¿Esta vacunado/a?"/>  
  
 <RadioGroup  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginBottom="10dp">  
 <RadioButton  
 android:id="@+id/rbVacunaSAnimalSubir"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_marginRight="10dp"  
 android:textSize="18dp"  
 android:text="Si" />  
 <RadioButton  
 android:id="@+id/rbVacunaNAnimalSubir"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:textSize="18dp"  
 android:text="No" />  
 </RadioGroup>  
  
 <TextView  
 android:id="@+id/txtChipAnimalSubir"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="26dp"  
 android:textColor="#212020"  
 android:text="¿Tiene chip?"/>  
  
 <RadioGroup  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginBottom="10dp">  
 <RadioButton  
 android:id="@+id/rbChipSAnimalSubir"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_marginRight="10dp"  
 android:textSize="18dp"  
 android:text="Si" />  
 <RadioButton  
 android:id="@+id/rbChipNAnimalSubir"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:textSize="18dp"  
 android:text="No" />  
 </RadioGroup>  
  
 <Button  
 android:id="@+id/btnSubirAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Subir animal"  
 android:background="@drawable/redondeadobutton"  
 android:textSize="20dp"  
 android:textColor="#ffffff"  
 android:onClick="subirNuevoAnimal"/>  
  
 </LinearLayout>  
  
 </ScrollView>  
  
</LinearLayout>

DI56. Content\_tuanimal.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.adoptales.rubenpalomo.adoptales.tuanimal"  
 android:background="#d6d6d6">  
  
 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.adoptales.rubenpalomo.adoptales.tuanimal"  
 tools:showIn="@layout/app\_bar\_tuanimal"  
 android:orientation="vertical"  
 android:layout\_marginBottom="60dp">  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/imgAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:src="@mipmap/ic\_launcher"  
 android:layout\_marginTop="50dp"/>  
  
 <TextView  
 android:id="@+id/txtNombreAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:text="Ruben Palomo"  
 android:textSize="50dp"  
 android:textStyle="bold"  
 android:textColor="#151414"  
 android:background="#c94db6ac"/>  
 </RelativeLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <LinearLayout  
 android:id="@+id/ly1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:orientation="vertical"  
 android:background="@drawable/datosanimalfondo">  
 <TextView  
 android:id="@+id/txtGeneroTu"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="..."  
 android:gravity="center"  
 android:textSize="25dp"  
 android:textColor="#000"/>  
 <ImageView  
 android:id="@+id/imgGenero"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:src="@drawable/gender"  
 android:padding="10dp"/>  
 </LinearLayout>  
 <LinearLayout  
 android:id="@+id/ly2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:orientation="vertical"  
 android:background="@drawable/datosanimalfondo">  
 <TextView  
 android:id="@+id/txtEdadTu"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="..."  
 android:gravity="center"  
 android:textSize="25dp"  
 android:textColor="#000"/>  
 <ImageView  
 android:id="@+id/imgAnos"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:src="@drawable/balloon"  
 android:padding="10dp"/>  
 </LinearLayout>  
 <LinearLayout  
 android:id="@+id/ly3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:orientation="vertical"  
 android:background="@drawable/datosanimalfondo">  
 <TextView  
 android:id="@+id/txtVacunaTu"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="..."  
 android:gravity="center"  
 android:textSize="25dp"  
 android:textColor="#000"/>  
 <ImageView  
 android:id="@+id/imgVacuna"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:padding="10dp"  
 android:src="@drawable/vaccine"  
 />  
 </LinearLayout>  
 <LinearLayout  
 android:id="@+id/ly4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:orientation="vertical"  
 android:background="@drawable/datosanimalfondo">  
 <TextView  
 android:id="@+id/txtChipTu"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="..."  
 android:gravity="center"  
 android:textSize="25dp"  
 android:textColor="#000"/>  
 <ImageView  
 android:id="@+id/imgChip"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:src="@drawable/chip"  
 android:padding="10dp"/>  
 </LinearLayout>  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:background="@drawable/redondeadoenviarcorreo"  
 android:gravity="center"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:padding="15dp">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <TextView  
 android:id="@+id/txtInformacion"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Informacion: "  
 android:textSize="30dp"  
 android:textColor="#151414"/>  
  
 <TextView  
 android:id="@+id/txtEmailAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=" hola@gmail.com"  
 android:textSize="20dp"  
 android:textColor="#212020"/>  
  
 <TextView  
 android:id="@+id/txtNumeroAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=" 6666666666"  
 android:textSize="20dp"  
 android:textColor="#212020"/>  
 </LinearLayout>  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:background="@drawable/redondeadoenviarcorreo"  
 android:gravity="center"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:padding="15dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <TextView  
 android:id="@+id/txtInfoAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Datos: "  
 android:textSize="30dp"  
 android:textColor="#151414"/>  
 <TextView  
 android:id="@+id/txtTipoAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Tipo"  
 android:textSize="20dp"  
 android:textColor="#212020"/>  
 <TextView  
 android:id="@+id/txtDescripcionAnimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Descripcion..."  
 android:textSize="18dp"  
 android:textColor="#151414"/>  
 </LinearLayout>  
 </LinearLayout>  
 </LinearLayout>  
 </ScrollView>  
 </LinearLayout>  
 <android.support.design.widget.BottomNavigationView  
 android:id="@+id/navigation"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="0dp"  
 android:layout\_marginStart="0dp"  
 android:background="#bf44a198"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:menu="@menu/navigationtuanimal"/>  
</android.support.constraint.ConstraintLayout>

DI57. listaanimales.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/redondeado">  
  
 <!-- Email label -->  
 <ImageView  
 android:id="@+id/img"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:background="#ffffff"  
 android:gravity="center"/>  
  
 <!-- Name Label -->  
 <TextView  
 android:id="@+id/nombreLista"  
 android:layout\_width="match\_parent"  
 android:layout\_height="45dp"  
 android:text="Name"  
 android:textSize="30dp"  
 android:textColor="#ffffff"  
 android:textStyle="bold|italic"  
 android:fontFamily="monospace"  
 android:gravity="center"  
 android:layout\_marginBottom="10dp"/>  
  
</LinearLayout>

DI58. listarefugio.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/redondeadobutton">  
  
 <!-- Email label -->  
 <ImageView  
 android:id="@+id/img"  
 android:layout\_width="match\_parent"  
 android:layout\_height="125dp"  
 android:textColor="#000000"  
 android:textSize="17dp"  
 android:fontFamily="monospace"  
 android:text="Email"  
 android:background="#ffffff"  
 android:gravity="center"/>  
  
 <!-- Name Label -->  
 <TextView  
 android:id="@+id/nombreListaRefugios"  
 android:layout\_width="match\_parent"  
 android:layout\_height="45dp"  
 android:text="Name"  
 android:textSize="30dp"  
 android:textColor="#ffffff"  
 android:gravity="center"  
 android:layout\_marginBottom="15dp"/>  
</LinearLayout>

DI59. Nav\_header\_oficial.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/nav\_header\_height"  
 android:background="@drawable/side\_nav\_bar"  
 android:gravity="bottom"  
 android:orientation="vertical"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="194dp"  
 android:layout\_height="76dp"  
 android:paddingTop="@dimen/nav\_header\_vertical\_spacing"  
 app:srcCompat="@drawable/icono" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="@dimen/nav\_header\_vertical\_spacing"  
 android:text="ADOPTALES"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="adoptalescontacto@gmail.com" />  
  
</LinearLayout>

DI60. Splashscreen.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:layout\_gravity="center"  
android:gravity="center"  
android:background="#4db6ac">  
  
<ImageView  
 android:layout\_width="257dp"  
 android:layout\_height="353dp"  
 android:layout\_gravity="center"  
 android:adjustViewBounds="false"  
 android:cropToPadding="false"  
 android:src="@drawable/icono" />  
  
</LinearLayout>

DI60. Activity\_pagina\_conregistro\_drawer.xml

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 >  
  
 <group android:checkableBehavior="single">  
 <item  
 android:id="@+id/nav\_inicio"  
 android:icon="@drawable/incio"  
 android:title="Inicio" />  
 <item  
 android:id="@+id/nav\_cuenta"  
 android:icon="@drawable/tucuenta"  
 android:title="Tu cuenta" />  
 <item  
 android:id="@+id/nav\_refugios"  
 android:icon="@drawable/refugio"  
 android:title="Refugios" />  
 <item  
 android:id="@+id/nav\_cerrarsesion"  
 android:icon="@drawable/cerrarsesion"  
 android:title="Cerrar Sesion" />  
 </group>  
  
 <item android:title="Comunicacion">  
 <menu>  
 <item  
 android:id="@+id/nav\_contacto"  
 android:icon="@drawable/contaco"  
 android:title="Contacto" />  
 <item  
 android:id="@+id/nav\_info"  
 android:icon="@drawable/informacion"  
 android:title="Sobre nosotros" />  
 <item  
 android:id="@+id/nav\_compartir"  
 android:icon="@drawable/ic\_menu\_share"  
 android:title="Compartir" />  
 </menu>  
 </item>  
  
</menu>

DI61. Activity\_pagina\_sinregistro\_drawer.xml

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <group android:checkableBehavior="single">  
 <item  
 android:id="@+id/nav\_inicio"  
 android:icon="@drawable/incio"  
 android:title="Inicio" />  
 <item  
 android:id="@+id/nav\_inicioSesion"  
 android:icon="@drawable/inciosesion"  
 android:title="Iniciar Sesion" />  
 <item  
 android:id="@+id/nav\_registro"  
 android:icon="@drawable/registro"  
 android:title="Registrarse" />  
 <item  
 android:id="@+id/nav\_refugios"  
 android:icon="@drawable/refugio"  
 android:title="Refugios" />  
 </group>  
  
 <item android:title="Comunicacion">  
 <menu>  
 <item  
 android:id="@+id/nav\_contacto"  
 android:icon="@drawable/contaco"  
 android:title="Contacto" />  
 <item  
 android:id="@+id/nav\_info"  
 android:icon="@drawable/informacion"  
 android:title="Sobre nosotros" />  
 <item  
 android:id="@+id/nav\_compartir"  
 android:icon="@drawable/ic\_menu\_share"  
 android:title="Compartir" />  
 </menu>  
 </item>  
  
</menu>

DI62. Navigationanimal.xml

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item  
 android:id="@+id/navigation\_llamar"  
 android:icon="@drawable/callphone"  
 android:title="Llamar" />  
  
 <item  
 android:id="@+id/navigation\_maps"  
 android:icon="@drawable/maps"  
 android:title="Localizacion"/>  
  
</menu>

DI63. Navigationtuanimal.xml

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item  
 android:id="@+id/navigation\_llamar"  
 android:icon="@drawable/delete"  
 android:title="Borrar" />  
  
 <item  
 android:id="@+id/navigation\_maps"  
 android:icon="@drawable/home"  
 android:title="Adoptado"/>  
  
</menu>

DI64. Pagina\_menu.xml

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
 <item  
 android:id="@+id/action\_ayuda"  
 android:orderInCategory="100"  
 android:title="Ayuda"  
 app:showAsAction="never"/>  
 <item  
 android:id="@+id/action\_info"  
 android:orderInCategory="100"  
 android:title="Sobre nosotros"  
 app:showAsAction="never"/>  
 <item  
 android:id="@+id/action\_contacto"  
 android:orderInCategory="100"  
 android:title="Contacto"  
 app:showAsAction="never"/>  
</menu>

DI65. Colors.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#0a8f7f</color>  
 <color name="colorPrimaryDark">#303F9F</color>  
 <color name="colorAccent">#FF4081</color>  
</resources>

DI66. Dimens.xml

<resources>  
 <!-- Default screen margins, per the Android Design guidelines. -->  
 <dimen name="activity\_horizontal\_margin">16dp</dimen>  
 <dimen name="activity\_vertical\_margin">16dp</dimen>  
 <dimen name="nav\_header\_vertical\_spacing">8dp</dimen>  
 <dimen name="nav\_header\_height">176dp</dimen>  
 <dimen name="fab\_margin">16dp</dimen>  
</resources>

DI67. Strings.xml

<resources>  
 <string name="app\_name">Adoptales</string>  
  
 <string name="navigation\_drawer\_open">Open navigation drawer</string>  
 <string name="navigation\_drawer\_close">Close navigation drawer</string>  
  
 <string name="action\_settings">Settings</string>  
  
 <string-array name="tipoanimal">  
 <item>Perro</item>  
 <item>Gato</item>  
 <item>Pajaro</item>  
 <item>Pez</item>  
 </string-array>  
  
 <string-array name="meses">  
 <item>Enero</item>  
 <item>Febrero</item>  
 <item>Marzo</item>  
 <item>Abril</item>  
 <item>Mayo</item>  
 <item>Junio</item>  
 <item>Julio</item>  
 <item>Agosto</item>  
 <item>Septiembre</item>  
 <item>Octubre</item>  
 <item>Noviembre</item>  
 <item>Diciembre</item>  
 </string-array>  
  
 <string-array name="dias">  
 <item>1</item>  
 <item>2</item>  
 <item>3</item>  
 <item>4</item>  
 <item>5</item>  
 <item>6</item>  
 <item>7</item>  
 <item>8</item>  
 <item>9</item>  
 <item>10</item>  
 <item>11</item>  
 <item>12</item>  
 <item>13</item>  
 <item>14</item>  
 <item>15</item>  
 <item>16</item>  
 <item>17</item>  
 <item>18</item>  
 <item>19</item>  
 <item>20</item>  
 <item>21</item>  
 <item>22</item>  
 <item>23</item>  
 <item>24</item>  
 <item>25</item>  
 <item>26</item>  
 <item>27</item>  
 <item>28</item>  
 <item>29</item>  
 <item>30</item>  
 <item>31</item>  
 </string-array>  
</resources>

DI68. Styles1.xml

<resources>  
  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
 <style name="AppTheme.NoActionBar">  
 <item name="windowActionBar">false</item>  
 <item name="windowNoTitle">true</item>  
 </style>  
  
 <style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar" />  
  
 <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />  
  
</resources>

DI69. Styles2.xml

<resources>  
  
 <style name="AppTheme.NoActionBar">  
 <item name="windowActionBar">false</item>  
 <item name="windowNoTitle">true</item>  
 <item name="android:statusBarColor">@android:color/transparent</item>  
 </style>  
</resources>

1. IMÁGENES UTILIZADAS.

|  |  |  |
| --- | --- | --- |
| O:\Adoptales\iconos\ayuda.png | O:\Adoptales\iconos\refugio.png | O:\Adoptales\app\src\main\res\drawable-v24\callphone.png |
| O:\Adoptales\iconos\cerrarsesion.png | O:\Adoptales\iconos\registro.png | O:\Adoptales\app\src\main\res\drawable-v24\chip.png |
| O:\Adoptales\iconos\contaco.png | O:\Adoptales\iconos\tucuenta.png | O:\Adoptales\app\src\main\res\drawable-v24\delete.png |
| O:\Adoptales\iconos\incio.png | O:\Adoptales\app\src\main\res\drawable\anadir.png | O:\Adoptales\app\src\main\res\drawable-v24\gender.png |
| O:\Adoptales\iconos\inciosesion.png | O:\Adoptales\app\src\main\res\drawable-v24\balloon.png | O:\Adoptales\app\src\main\res\drawable-v24\home.png |
| O:\Adoptales\iconos\informacion.png | O:\Adoptales\app\src\main\res\drawable-v24\vaccine.png | O:\Adoptales\app\src\main\res\drawable-v24\maps.png |

1. ASEGURAMIENTO DE LA CALIDAD.
2. EXPERIMENTACIÓN/ PRUEBAS/ VALIDACIÓN.
3. EVALUACIÓN/ DISCUSIÓN.

Esta aplicación ha sido desarrollada en Android Studio con una conexión a Base de Datos MySQL. Mi evaluación sobre mi trabajo en esta aplicación es que sinceramente para el poco tiempo que he tenido para desarrollarla ha salido bastante bien con una buena planificación de tiempo. Intentando quitarme el mayor peso de la aplicación antes de comenzar las practicas dejando el diseño y pequeños modulitos para el final. Los problemas que me he encontrado al desarrollarla ha sido el controlar las fotos y un buen diseño que es de lo más importante. También se han utilizado librerías externas para diferentes casos como enviar emails y recuperar las fotos del servidor.

1. CONCLUSIONES.

Mi conclusión sobre el trabajo desarrollado en esta aplicación que se han cumplido todos o casi todos los objetivos propuestos claramente me hubiera gustado hacer más cosas y hacer una aplicación más completa, aunque ahora no me haya dado tiempo voy a seguir desarrollándola y lanzándola al mercado. La calidad de esta app es buena, yo creo que cumple las expectativas y las necesidades de los usuarios y de las mascotas.

1. TRABAJOS FUTUROS

TF1. Mejorar diseño de datos.

TF2. Añadir un grupo de favoritos.

TF3. Añadir modificación de datos.

TF4. Mejorar el icono.

TF5. Prepararlo para Apple.

TF6. Añadir un chat para que los usuarios puedan hablar entre ellos para adoptar los animales.

1. GUÍA DE SOLUCIÓN DE ERRORES.
2. GLOSARIO DE TÉRMINOS.

**1. MySql**: Un sistema de base de datos.

**2. SqLite:** Un sistema de base de datos interno en la app.

**3. Registro:** Un sistema que guarda un registro de datos, una pequeña cantidad de datos.

**4. Android Studio:** Es un entorno de desarrollo

**5. Base de datos:** Guarda la información.

**6. XML:** Lenguaje de programación que con el que se crea el diseño de las pantallas.

**7. Java:** Lenguaje de programación con el que se le da funcionamiento a las pantallas.

**8. Metodología en cascada:** Es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

**9. Web Service:** Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones

1. BIBLIOGRAFÍA.

**Página Web: Android Studio FAQs - Tutoriales Android Studio.**

<https://androidstudiofaqs.com>

**Página Web: DaFont - Descargar fuentes.**

<https://www.dafont.com/es/>

**Página Web: 867,500 vector icons grouped in 16,941 packs.**

<https://www.flaticon.com>

**Página Web: W3Schools Online Web Tutorials.**

<https://www.w3schools.com>

**Página Web: Mostrar/cargar imagen externa en una aplicación Android.**

<http://jonsegador.com/2010/03/mostrarcargar-imagen-externa-en-unaaplicacion-android/>

**Página Web: Tutorial Para Crear Un GridView En Android.**

<http://www.hermosaprogramacion.com/2015/07/tutorial-para-crear-un-gridviewen-android/>

**Página Web: Adopción en Animales.**

<https://es.wikipedia.org/wiki/Adopción_en_animales>

**Página Web: ¿Por qué es importante adoptar?**

<http://www.cuidandotumascota.com/blog/posts/por-que-es-importante-adoptar/>

1. APENDICES

**Guía de Instalación:**

**Forma 1: APK**

1. Cargar o descargar la APK en tu móvil.

2. Pinchar en la APK.

3. Permitir aplicaciones desconocidas.

4. Esperar a que se instale.

5. Ya puedes adoptar tu animalito.

**Forma 2: Google Play**

1. Ir a la página de google play de la aplicación.

2. Pinchar en la Instalar.

3. Esperar a que se instale.

4. Ya puedes adoptar tu animalito.

**Manual de Usuario:**

**VIDEO**