Memoria PDF proyecto Formulario Android.

Pantalla 1 (MainActivity)

Una vez iniciemos la aplicación, nos saldrá primero de todo un alert el cual nos ofrecerá iniciar sesión para poder acceder a determinadas funciones que estarán bloqueadas por defecto.

Esta primera pantalla dispone de dos CheckBoxes, con los cuales añadiremos al envío la opción de enviar el paquete con una caja de regalo o una tarjeta dedicada, dispondrá también de un EditText para el peso, un spinner para seleccionar la zona a la cual sera enviada el paquete (estas zonas son definidas en una clase aparte la cual detallare mas adelante) un RadioGroup para la urgencia del pedido por ultimo un botón para enviar los datos, todos estos objetos son definidos en el XML Res/layout/activity_main.xml

Entrando a la lógica de aplicación punto por punto, antes del OnCreate, inicializamos un array de Destinos donde incorporo los objetos con los cuales se rellenara el spinner.

```
//array con objetos de tipo Dest para el spinner
private Dest[] objetosSpinner = new Dest[]{
            new Dest("A","Asia/Oceania",R.drawable.asia, "30"),
            new Dest("B","America/Africa",R.drawable.america,"20"),
            new Dest("C","Europa",R.drawable.europa,"10")
};
```

Una de las primeras cosas que se hago en la MainActivity, es incorporar un bundle, un TextView para el nombre de usuario, esto es, cuando iniciamos la aplicación como dije te preguntara si quieres iniciar sesión o no, este bundle y su comprobación de si esta vacío o no, es para una vez iniciada sesión y después de navegar entre las pantallas, guarde el usuario iniciado atrevas de la Pantalla Login(dedicado al registro e inicio de sesión) y así poder desbloquear funcionalidades.

```
//este textview indica el usuario activo y se muestra junto al logo de la empresa
nombreUsuario=(TextView)findViewById(R.id.nombreUsuario);
//con este if else y este bundle compruebo si hay una sesion iniciada y saco un mensaje.
Bundle bundle;
if(( bundle = getIntent().getExtras())==null){
      /*Toast.makeText(MainActivity.this, "No estas inciado como ningun usuario, puede hacerlo
desde menu -> Registrar, para acceder a nuevas funciones", Toast.LENGTH_SHORT).show();
      nombreUsuario.setText("Anonimo");*/
      new AlertDialog.Builder(this)
                  .setTitle("Sesion no iniciada")
                  .setMessage("No estas inciado como ningun usuario, puede hacerlo desde
menu,Registrar, para acceder a nuevas funciones")
                  .setPositiveButton("iniciar sesion", new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int which) {
                              Intent activityIntent2 = new Intent(MainActivity.this,
PantallaLogin.class);
                              startActivity(activityIntent2);
                        }
                  })
                  .setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int which) {
                              nombreUsuario.setText("Anonimo");
                              Toast.makeText(MainActivity.this, "Iniciado como Anonimo",
Toast.LENGTH_SHORT).show();
                        }
                  })
                  .setIcon(android.R.drawable.ic_dialog_alert)
                  .show();
```

```
}
else{
      bundle = getIntent().getExtras();
      String nombreU = bundle.getString("Nombre");
      nombreUsuario.setText(nombreU);
}
```

Continuando con el codigo, inicializo los objetos de la pantalla (TextView etc) con sus métodos escuchadores correspondientes, los RadioButton y los CheckBoxes no son altamente importantes, ya que lo único que realizan es modificar un String en función de los seleccionado para posteriormente pasarlo a un bundle, sin embargo en la inicialización del Spinner es necesario crear un adaptador especifico ya que uso objetos de tipo Dest (clase propia que guarda los destinos) mas adelante describiré la creación de ese adaptador.

```
cajaRegalo = (CheckBox) findViewById(R.id.cajaRegalo);
conTarjetaDedicada = (CheckBox) findViewById(R.id.conTarjetaDedicada);
Peso = (EditText) findViewById(R.id.Peso);
MiAdaptador adaptador = new MiAdaptador(this);
miSpinner = (Spinner) findViewById(R.id.spin);
miSpinner.setAdapter(adaptador);
radioGroup = (RadioGroup) findViewById(R.id.grupoRadio);
radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
      @Override
      public void onCheckedChanged(RadioGroup group, int checkedId) {
            if (R.id.rb1 == radioGroup.getCheckedRadioButtonId()) {
                  Urgencia = "Tarifa Normal";
            } else {
                  Urgencia = "Tarifa Urgente";
                  precioExtra = precioExtra+(pr*30)/100;
            }
      }
});
cajaRegalo.setOnCheckedChangeListener(new CheckBox.OnCheckedChangeListener() {
      public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
            if (cajaRegalo.isChecked()) {
                  preferenciasDelEnvio = "con caja regalo";
            } else {
                  preferenciasDelEnvio = "Ninguno";

            }
      }
});
conTarjetaDedicada.setOnCheckedChangeListener(new CheckBox.OnCheckedChangeListener()
{
      public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
            if (conTarjetaDedicada.isChecked()) {
                  preferenciasDelEnvio = "Envuelto y con una tarjeta dedicatoria";
            } else {
                  preferenciasDelEnvio = "ninguno";
            }
      }
});
```

Después tenemos la inicialización del botón calcular y su metodo OnClickListener, este tiene una doble función, primero comprueba el valor del EntryText Peso, de forma que en función de su valor modifique el valor de precio extra para sumarlo al precio (el cual extraigo del spinner), Segundo crea tanto el Intent como el Bundle y rellena este con todas las variables que han ido recogiendo los valores de los campos de la pantalla, Zona, Continente, Precio e Imagen, hacen referencia a los atributos de los objetos de tipo Dest con los que relleno el spinner (apartir del arra inicial), PrecioExtra es una variable de tipo double con la cual suma, aquellos gastos aparte del precio fijo por zona, como por ejemplo la cantidad extra a pagar por el peso del paquete, precio total es la suma de todos los precios, para mayor claridad, urgencia del envío obtiene el valor del CheckBox seleccionado, y el nombre de usuario recoge el valor de TextView nombreUsuario (Rellenado por el bundle inicial a través de lo obtenido en la Pantalla Login).

Una vez el bundle es rellenado se pasa al Intent e inicio la Pantalla2.

```
 calcular = (Button) findViewById(R.id.btCal);
    calcular.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            if (Peso.getText().toString().isEmpty()) {
                Toast.makeText(getApplicationContext(), "Inserte el peso del paquete",
Toast.LENGTH_SHORT).show();
            } else {
                int peso = Integer.parseInt(Peso.getText().toString());
                if(peso<=5){
                    precioExtra = precioExtra + peso;
                }
                if(peso>=6 || peso<10){
                    precioExtra= precioExtra + peso*1.5;
                }
                if(peso>10){
                    precioExtra = precioExtra + peso*2;
                }
                Intent intent = new Intent(MainActivity.this, Pantalla2.class);
                Bundle bundle = new Bundle();
                bundle.putString("Zona", Zon);
                bundle.putString("Continente", Cont);
                bundle.putDouble("precio", pr);
                bundle.putDouble("PrecioExtra", precioExtra);
                bundle.putInt("Peso", peso);
                bundle.putDouble("PRECIO TOTAL (Precio + precio extra por tipo de envio",
pr + precioExtra);
                bundle.putString("urgencia del envio", Urgencia);
                bundle.putInt("IMAGEN", img);
                bundle.putString("Usuario",nombreUsuario.getText().toString());
                intent.putExtras(bundle);
                startActivity(intent);
            }
        }
    });
}//Fin de OnCreate
```

Para finalizar la MainActivity esta la creación del menu y la clase miAdaptador usada para rellenar el spinner, para la creación del menú lo inflo apartir del XML menu_main, dentro de la carpeta Res->menu

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

Para aplicar lógica de aplicación al menú realizo un switch case, en función de que ítem del menu sea seleccionado, estos, los identifico con la id asignada en el XML menu_main, por tanto este menú podrá pasar a las pantallas :
1. Dibujar (esta pantalla realizara atreves de un canvas un dibujo sencillo de una muñeca.
2. Una opción de Acerca_De la cual mostrara información sobre el creador.
3.En esta tercera opción podremos saltar a la pantalla login para registrarnos o iniciar sesion, si ya estamos iniciados como un usuario el botón registrar de la pantalla login sera invisible, por eso hay un bundle que recoge el valor del TextView nombreUsuario.
4.La cuarta opción es una forma sencilla de cerrar sesión, inicializando el TextView del usuario a Anonimo.
5.Con la quinta pasaremos a una pantalla para poder leer las noticias.

```
public boolean onOptionsItemSelected(MenuItem ítem) {
    switch (item.getItemId()) {
        case R.id.Opc1:
            Intent activityIntent = new Intent(this, Dibujar.class);
            startActivity(activityIntent);
            return true;
        case R.id.Opc2:
            Toast.makeText(getApplicationContext(), "Diego de Belda Calvo",
Toast.LENGTH_SHORT).show();
            return true;
        case R.id.Opc3:
            Bundle bundle = new Bundle();
            String usuarioyainiciado = nombreUsuario.getText().toString();
            bundle.putString("Usuario", usuarioyainiciado);
            Intent activityIntent2 = new Intent(this, PantallaLogin.class);
            activityIntent2.putExtras(bundle);
            startActivity(activityIntent2);
            return true;
        case R.id.Opc4:
            nombreUsuario.setText("Anonimo");
            return true;
        case R.id.Opc5:
            Intent activityIntent3 = new Intent(this, PantallaElPais.class);
            startActivity(activityIntent3);
            return true;
        default:
            return super.onOptionsItemSelected(ítem);
    }
}
```

Por ultimo esta la clase interna miAdaptador para crear el array adapter y así rellenar el spinner.

En esta clase se encuentran:

1. La declaración del contexto dentro de la clase.
2. El constructor que usamos en el spinner este constructor, tiene como parámetros, un contexto, un XML, en este caso R/layout/ítem_spinner.xml y el array de objetos con el que rellenaremos el spinner.

3. El metodo View getDropDownView, este metodo sirve para poder deslizar con el dedo el spinner.
4. Ya por ultimo tenemos el metod getView para inflar el spinner, aquí, inflamos el spinner con un layout inflater, haremos un convertView atrevas del inflater usando un xml con los items de spinner (R/layout/items_spinner.xml), Inicializamos tres TextViews (Zona, Continente y precio) dentro del spinner y un ImageView también dentro del spinner con la posición del array de objetos de tipo Dest usando los métodos get correspondientes, de la misma forma inicializamos las variables zon, cont, pr e img las cuales son las que paso al bundle en el botón calcular, por ultimo devuelve la vista.

```
class MiAdaptador extends ArrayAdapter<Dest> {
    Activity context;
    MiAdaptador(Activity context) {
        super(context, R.layout.items_spinner, objetosSpinner);
        this.context = context;
    }
    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        return getView(position, convertView, parent);
    }
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = getLayoutInflater();
        convertView = inflater.inflate(R.layout.items_spinner, null);
        TextView Zona = (TextView) convertView.findViewById(R.id.zona);
        Zona.setText(objetosSpinner[position].getZona());
        TextView Continente = (TextView) convertView.findViewById(R.id.continente);
        Continente.setText(objetosSpinner[position].getContinente());
        ImageView imagen = (ImageView) convertView.findViewById(R.id.Imagen);
        imagen.setImageResource(objetosSpinner[position].getImagen());
        TextView precio = (TextView) convertView.findViewById(R.id.precio);
        precio.setText(String.valueOf(objetosSpinner[position].getPrecio()));
        Zon = objetosSpinner[position].getZona();
        Cont = objetosSpinner[position].getContinente();
        img = objetosSpinner[position].getImagen();
        pr = Double.parseDouble(objetosSpinner[position].getPrecio());
        return convertView;
    }
}
```

Activity_Main.xml (activity_main.xml)

En este XML, quedan definidas todas las propiedades, de los objetos de la Pantalla 1.

Primero de todo, tenemos un LinearLayout sobre el cual desarrollaremos todo de todos los atributos que tiene lo unico realmente interesante es toolds:context=".MainActivity" para indicar el contexto del XML, y la propiedad android:background para añadirle una opción de fondo, en mi caso use un degradado el cual tengo almacenado en Drawable.

Dentro encontramos:

1. Un LinearLayout para añadir El nombre de la apicación, el logo de la empresa y el nombre del usuario iniciado.
2. Otro LinearLayout donde se encuentra el spinner (a diferencia de la lógica de aplición el diseño de un spinner en el xml es realmente sencillo)
3. Un tercer LinearLayout para los CheckBoxes.
4. Un cuarto para los RadioButton (los cuales están dentro de un RadioGroup)
5. Por último uno más para el EditText del peso y el botón calcular.

De esta forma, al tener varios LinearLayout dentro de uno, conseguimos separar los contenidos para una vista mas ordenada y limpia.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity"
    android:orientation="vertical"
    android:divider="?android:dividerHorizontal"
    android:showDividers="middle"
    android:weightSum="1"
    android:background="@drawable/fondo">
    <LinearLayout
        android:layout_width="350dp"
        android:layout_height="wrap_content"
        android:weightSum="1"
       >
        <TextView
            android:layout_width="130dp"
            android:layout_height="20dp"
            android:textStyle="bold"
            android:textSize="20dp"
            android:text="Envios D.De Belda"
            android:id="@+id/textView"
            android:textColor="#000000"
            android:layout_marginBottom="30dp"
            android:layout_marginTop="10dp"
            android:layout_weight="0.58" />
        <ImageView
            android:layout_width="50dp"
            android:layout_height="match_parent"
            android:id="@+id/imageView"
```

```
            android:background="@drawable/logo"
            android:paddingRight="10dp" />
        <TextView
            android:layout_width="60dp"
            android:layout_height="50dp"
            android:text="      Usuario: "
            android:id="@+id/nombreUsuario"/>
    </LinearLayout>
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <Spinner
            android:id="@+id/spin"
            android:layout_height="wrap_content"
            android:layout_width="350px"
            android:layout_marginRight="20dp" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <CheckBox
            android:layout_width="101dp"
            android:layout_height="wrap_content"
            android:text="caja regalo"
            android:typeface="sans"
            android:id="@+id/cajaRegalo"
            android:layout_marginRight="15dp"/>
        <CheckBox
            android:layout_width="104dp"
            android:layout_height="wrap_content"
            android:text="con tarjeta dedicada"
            android:typeface="sans"
            android:id="@+id/conTarjetaDedicada"
            android:layout_marginBottom="20dp"/>
    </LinearLayout>
    <RadioGroup
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:gravity="center"
        android:id="@+id/grupoRadio">
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:typeface="sans"
            android:gravity="center"
            android:text="Tarifa Urgente"
            android:id="@+id/rb2"
            />
        <RadioButton
            android:layout_width="121dp"
            android:layout_height="wrap_content"
            android:typeface="sans"
            android:gravity="center"
            android:text="Tarifa Normal"
            android:id="@+id/rb1"
            android:checked="true"/>
```

```
    </RadioGroup>
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <EditText
            android:layout_width="165dp"
            android:layout_height="wrap_content"
            android:hint="Peso"
            android:inputType="numberDecimal"
            android:id="@+id/Peso"
            />
        <Button
            android:layout_width="165dp"
            android:layout_height="wrap_content"
            android:text="Calcular "
            android:typeface="sans"
            android:id="@+id/btCal"
            android:layout_gravity="right" />
    </LinearLayout>
</LinearLayout>
```

Pantalla Login (pantalla de inicio de sesión o de registro)

En esta pantalla vamos a tener dos EditText, uno para usuario y otro para la contraseña, dos botones Inicio de Sesion y Registrar (este ultimo, estará deshabilitado en caso de que lleguemos a esta pantalla ya iniciados como un usuario) , un objeto de la clase Usuarios (esta clase solo contiene dos String, uno para usuario y otro para contraseña y los getter and setter correspondientes), dos String que usare en los EditText, un objeto SQLiteDatabase para la base de datos de usuarios, y un SQLiteHelper para trabajar con la susodicha base de datos.

```
EditText usuario;
EditText contraseña;
Context context;
Usuarios usuarios;
SQLiteHelper cliBDh;
SQLiteDatabase db;
String auxUsuario;
String auxContraseña;
```

Ya dentro del onCreate, asigno el XML correspondiente a esta pantalla (R/layout/pantalla_registro), inicializo los EditText de usuario y contraseña a partir del XML anterior, y declaro dos TextView los cuales están ocultos y solo saldrán en caso de error, uno cuando intente iniciar sesion estando ya iniciado y otro cuando el usuario y contraseña no coincidan.

```
super.onCreate(savedInstanceState);
setContentView(R.layout.pantalla_registro);
usuario= (EditText)findViewById(R.id.editText);
contraseña = (EditText)findViewById(R.id.editText2);
final TextView mensajeError =(TextView)findViewById(R.id.textView3);
final TextView mensajeErrorDos = (TextView)findViewById(R.id.textView4);
```

Luego tenemos la declaración de los botones RegistrarUsuario e IniciarSesion, empezando por registrar, dentro del onClick tenemos que recoge el valor de los EditText de usuario y contraseña y los pasa a las variables auxUsuario y auxContraseña, en caso de que ambos No esten vacios, abrira la base de datos DBClientes en modo escritura y llamara al metodo de insertar usuarios (perteneciente a la clase SQLiteHelper), luego se cierran el helper y la base de datos .

```
Button RegistrarUsuario = (Button)findViewById(R.id.button);
Button IniciarSesion = (Button)findViewById(R.id.button2);
RegistrarUsuario.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        auxUsuario = usuario.getText().toString();
        auxContraseña = contraseña.getText().toString();
        if( auxUsuario!=null && auxContraseña!=null){
        usuarios = new Usuarios(auxUsuario, auxContraseña);
        cliBDh=new SQLiteHelper(PantallaLogin.this, "DBClientes", null, 1);
        db=cliBDh.getWritableDatabase();
        cliBDh.insertarBDUsuario(db,PantallaLogin.this,usuarios);
            cliBDh.close();
            db.close();
        }
    }
});
```

Pasando con iniciar sesion recogemos los EditText de la misma forma que en el registrar, en caso de que ambos No estén vacíos, creo un objeto de la clase Usuarios usando como atributos lo recogido de los EditText, después abro la base de datos en modo lectura, y relleno un array de Usuarios con el metodo listarUsuarios perteneciente a la clase SQLiteHelper, el cual devuelve una query apartir de una base de datos, con un for compruebo que el usuario y contraseña exista y coincida en el array de usuarios, si coincide, sacara un Toast con el usuario iniciado, lo meterá en un bundle y lo pasara a la pantalla 1 (MainActivity), en caso de que no coincida haremos visible el mensaje de error correspondiente al fallo de inicio de sesión y cerraremos la conexión con la base de datos.

Antes de terminar la clase, recojo un bundle para comprobar si ya estas iniciado como algún usuario y así deshabilitar el botón iniciar sesión y mostrar el mensaje de error.

```
IniciarSesion.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        auxUsuario = usuario.getText().toString();
        auxContraseña = contraseña.getText().toString();
        if( auxUsuario!=null && auxContraseña!=null){
        Usuarios nuevo=new  Usuarios (auxUsuario,auxContraseña);
            cliBDh=new SQLiteHelper(PantallaLogin.this, "DBClientes", null, 1);
            db=cliBDh.getReadableDatabase();
            Usuarios[] array=cliBDh.listarUsuarios(db);
            for (int i=0; i<array.length; i++){
                //Toast.makeText(PantallaLogin.this, ""+array[i].getNombre(),
Toast.LENGTH_SHORT).show();
                if(nuevo.getNombre().equals(array[i].getNombre()) &&
nuevo.getPassword().equals(array[i].getPassword())){
                    Toast.makeText(PantallaLogin.this, "Sesion iniciada como usuario
"+nuevo.getNombre()+" Volviendo a la pantalla 1", Toast.LENGTH_LONG).show();
                    Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                    String jhu=array[i].getNombre();
                    Bundle bundle = new Bundle();
                    bundle.putString("Nombre", jhu);
                    intent.putExtras(bundle);
                    startActivity(intent);
                }
            }
            mensajeError.setVisibility(View.VISIBLE);
            cliBDh.close();
            db.close();
        }
    }
});
Bundle bundle = getIntent().getExtras();
String comprobar = bundle.getString("Usuario");
if(comprobar.equals("Anonimo")==false){
    mensajeErrorDos.setText(mensajeErrorDos.getText().toString()+" como "+comprobar);
    IniciarSesion.setVisibility(View.INVISIBLE);
    mensajeErrorDos.setVisibility(View.VISIBLE);
}
```

Pantalla de Login XML (pantalla_registro.xml)

En esta pantalla tengo un LinearLayout en el cual se encuentran, El TextView con el nombre de la pantalla, los EditText de usuarios y contraseña, después hay dos LinearLayout mas , uno contendrá los botones de registrar e iniciar y otro que contendrá los TextView de errores, puestos en invisible y rojos.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="  Ingrese su usuario"
        android:id="@+id/textView2"
        android:layout_gravity="center_horizontal"
        android:layout_weight="0.19"
        android:textColor="#10efb4"
        android:textSize="40dp" />
    <EditText
        android:layout_width="286dp"
        android:layout_height="wrap_content"
        android:id="@+id/editText"
        android:hint="Usuario"
        android:layout_gravity="bottom" />
    <EditText
        android:layout_width="285dp"
        android:layout_height="wrap_content"
        android:id="@+id/editText2"
        android:hint="Contraseña"
        android:password="true" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="210dp"
        android:orientation="horizontal"
        android:weightSum="1">
        <Button
            android:layout_width="149dp"
            android:layout_height="wrap_content"
            android:text="Registrar"
            android:id="@+id/button" />
        <Button
            android:layout_width="146dp"
            android:layout_height="wrap_content"
            android:text="Iniciar sesion"
            android:id="@+id/button2"
            android:layout_weight="0.31" />
    </LinearLayout>
    <TextView
        android:layout_width="257dp"
        android:layout_height="wrap_content"
        android:id="@+id/textView3"
        android:layout_weight="0.19"
        android:text="Usuario o contraseña incorrecto"
        android:textColor="#f90808"
```

```
            android:visibility="invisible" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView4"
            android:layout_weight="0.19"
            android:text="Ya has iniciado sesion."
            android:textColor="#f90808"
            android:visibility="invisible" />
</LinearLayout>
```

Clases Dest y Usuarios

Estas clases no extienden de AppCompatActivity, por lo que no serán pantallas si no clases java, en ellas encontraremos los atributos correspondientes de cada clase, sus getter y sus setter además de los constructores.

Dest

```
package com.example.diego.formulario;
class Dest{
private String Zona;
private String Continente;
private String precio;
private int imagen;
public Dest(String zona, String continente, int imagen, String precio ) {
            this.Zona=zona;
            this.Continente=continente;
            this.imagen = imagen;
            this.precio = precio;
            }
public String getZona() {
            return Zona;
            }
public String getContinente(){return Continente;}
            public int getImagen() {
                        return imagen;
            }
            public String getPrecio() {
            return precio;
            }
public void setZona(String zona) {
            this.Zona = zona;
}
            public void setContinente(String continente){this.Continente=continente;}
public void setImagen(int imagen) {this.imagen = imagen;}
            public void setPrecio(String precio) {
                        this.precio = precio;
            }
            }
```

Usuarios

```
package com.example.diego.formulario;
/**
 * Created by mati on 8/02/16.
 */
public class Usuarios {
      private String nombre;
      private String password;
      public String getNombre() {
            return nombre;
      }
      public String getPassword() {
            return password;
      }
      public void setNombre(String nombre) {
            this.nombre = nombre;
      }
```

```
    public void setPassword(String password) {
        this.password = password;
    }
    public Usuarios(String nombre, String password){
        this.nombre=nombre;
        this.password=password;
    }
}
```

Pantalla Dibujar

A esta clase podemos llegar a través de la opción Dibujar del menu de la primera pantalla, aquí, creo un objeto de la clase Dibujando llamada AreaDibujo y le asignare la view del XML correspondiente a Dibujar (R/layout/activity_dibujo.xml).

activity_dibujo.xml

En este XML, dentro de un LinearLayout, tenemos un view al cual le asignamos la clase Dibujando.

Clase Dibujando

En esta clase se encuentra toda la lógica para poder dibujar la figura, “simplemente” a través de un pincel, se van dibujando en un canvas las lineas necesarias para la figura.

Pantalla 2

La función de la pantalla 2 consiste en recoger el bundle pasado a través del botón calcular de la pantalla 1 y mostrar todo su contenido,todo, a partir del XML correspondiente (R/layout/activity_pantalla2.xml) para ello:

Dentro del onCreate inicializo todos los TextView (y un ImageView) todo, lo que hemos recogido del bundle.

```
final TextView viewZona = (TextView) findViewById(R.id.tzona);
final TextView viewContinente = (TextView) findViewById(R.id.tcontinente);
final TextView viewPrecio = (TextView) findViewById(R.id.tprecio);
final TextView viewPrecioExtra = (TextView) findViewById(R.id.tprecioextra);
final TextView viewPrecioTotal = (TextView) findViewById(R.id.tpreciototal);
final TextView viewPeso = (TextView)findViewById(R.id.tpeso);
final ImageView viewImagen = (ImageView) findViewById(R.id.timagen);
final TextView viewUsuario = (TextView)findViewById(R.id.tusuario);
```

Recojo el bundle y asigno cada cosa a su Text/Image View correspondiente.

```
Bundle bundle = getIntent().getExtras();
String Zona = bundle.getString("Zona");
String Continente = bundle.getString("Continente");
double precio = bundle.getDouble("precio");
double precioExtra = bundle.getDouble("PrecioExtra");
double precioFinal = precio + precioExtra;
int Peso = bundle.getInt("Peso");
double suma = bundle.getDouble("PRECIO TOTAL (Precio + precio extra por tipo de envio");
String urgencia = bundle.getString("urgencia del envio");
String usuario = bundle.getString("Usuario");

int img = bundle.getInt("IMAGEN");

String envio = bundle.getString("Envio");
```

```
viewZona.setText("Zona: "+Zona);
viewContinente.setText("Continente: "+Continente);
viewPrecio.setText(String.valueOf("Precio: "+precio));
viewPrecioExtra.setText(String.valueOf("Precio extra: " + precioExtra));
viewPrecioTotal.setText(String.valueOf("Precio TOTAL: " + precioFinal));
viewPeso.setText(String.valueOf("Peso del paquete: " + Peso));
viewImagen.setImageResource(img);
viewUsuario.setText(usuario);
```

Ahora, creo los escuchadores de los botones cargar en la base de datos y listar la base de datos, cargar, insertara en la base de datos, listar, va a cargar una nueva pantalla (pantalla3) para ver todos los pedidos, la inserción de los pedidos es muy parecida a la de usuarios, pero cambiando el metodo.

```
Button cargar = (Button)findViewById(R.id.Registrar);
destino = new Dest(Zona, Continente, img, ""+precioFinal);
cargar.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
          cliBDh=new SQLiteHelper(Pantalla2.this, "DBClientes", null, 1);
          db=cliBDh.getWritableDatabase();
          cliBDh.insertarBD(db,Pantalla2.this,destino);
      }
});
Button ver = (Button)findViewById(R.id.Visualizar);
ver.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
           Intent intent = new Intent(getApplicationContext(), pantalla3.class);
           startActivity(intent);
      }
});
```

Por ultimo hay una comprobación interesante, recogiendo del bundle el atributo usuario, si es Anonimo deshabilitaremos estos dos botones, haciéndolos invisibles.

```
if(usuario.equals("Anonimo")){
      cargar.setVisibility(View.INVISIBLE);
      ver.setVisibility(View.INVISIBLE);
}
```

Pantalla 3

Esta pantalla tiene cierto parecido con la MainActivity, su función sera rellenar un ListView a partir de todos los registros de una tabla de pedidos en la base de datos, todo, apartir de su XML (R/layout/pantalla3.xml) para ello:

Primero de todo creo las variables, un array de Dest para inflar la ListView, dos String para Zona y Continente, un doble de precio, y un int para la imagen.

El array de Dest lo relleno con el metodo listarBD el cual trabaja con el metodo listar de la clase SQLiteHelper, que me devuelve un array de los pedidos.

Luego creo la ListView y le asigno un adaptador a partir de la clase interna ListAdapter, este adaptador funciona muy parecido al del spinner de la MainActivity, solo que para el constructor en vez de contexto usaremos un activity, después el metodo getView es prácticamente idéntico.

```
public class pantalla3 extends Activity {
    Dest[] destino ;
    String Zon, Cont;
    double pr;
    int img;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.pantalla3);
        destino = listarBD(this);
        ListView lista =(ListView) findViewById(R.id.listView);
        ListAdapter adaptador = new ListAdapter(this);
        lista.setAdapter(adaptador);
    }
    public Dest[] listarBD(Context context){
        SQLiteHelper cliBDh = new SQLiteHelper(context, "DBClientes", null, 1);
        SQLiteDatabase db = cliBDh.getReadableDatabase();
        return cliBDh.listar(db);
    }
    public class ListAdapter extends ArrayAdapter {
        Activity main;
        public ListAdapter (Activity activity){
            super(activity,R.layout.itemlist, destino);
            main=activity;
        }
        @Override
        public View getView(int position, View convertView, ViewGroup parent) {
            LayoutInflater inflater = getLayoutInflater();
            convertView = inflater.inflate(R.layout.itemlist, null);
            TextView Zona = (TextView) convertView.findViewById(R.id.zona2);
            Zona.setText(destino[position].getZona());
            TextView Continente = (TextView) convertView.findViewById(R.id.continente2);
            Continente.setText(destino[position].getContinente());
            ImageView imagen = (ImageView) convertView.findViewById(R.id.Imagen2);
            imagen.setImageResource(destino[position].getImagen());
            TextView precio = (TextView) convertView.findViewById(R.id.precio2);
            precio.setText(String.valueOf(destino[position].getPrecio()));
            Zon = destino[position].getZona();
            Cont = destino[position].getContinente();
```

```
                img = destino[position].getImagen();
                pr = Double.parseDouble(destino[position].getPrecio());
                return convertView;
            }
        }
}
```

pantalla3.xml (R/layout/pantalla3.xml)

Este layout únicamente contiene un LinearLayout y dentro de el la ListView con la que Pantalla3 trabaja.

PantallaElPais

La pantalla de lectura del periódico del país y su xml correspondiente es igual que el que hicimos en clase corrigiendo algunos fallos como la colocación de los permisos en el manifest.

Clase SQLiteHelper

Esta clase sera la encagada de todo el trabajo relacionado con inserción, modificación, borrado y listado de la base de datos así como de la creación de las query.

Primera creo de tipo string todas las sentencias SQL.

```
String cadSQL = "CREATE TABLE Destinos if not EXISTS (id INTEGER PRIMARY KEY, zona
TEXT,continente TEXT, imagen INTEGER, precio TEXT )";
String cadSQLUsuarios = "CREATE TABLE Usuarios if not EXISTS (id INTEGER PRIMARY KEY,
nombre TEXT, password TEXT)";
String cadListar = "SELECT * FROM Destinos";
String cadListarUsuarios = "SELECT * FROM Usuarios";
```

Creo el constructor de la clase, con el cual creo el objeto cliBDh el cual uso en varias pantallas.

```
public SQLiteHelper(Context contexto, String nombre, CursorFactory almacen, int version){
    super(contexto, nombre, almacen, version);
}
```

El metodo insertarBD inserto los pedidos en la base de datos, en su tabla correspondiente

```
public void insertarBD( SQLiteDatabase db, Context context, Dest destino ){
    if(db!=null){
    db.execSQL("INSERT INTO Destinos (zona, continente, imagen, precio) " +
                "VALUES ('" + destino.getZona() + "', '"+destino.getContinente()+"',
'"+destino.getImagen()+"', '"+destino.getPrecio()+"')");
    db.close();
        Log.d("prueba", "todo bien");
    }
    else{
        Toast.makeText(context, "Error al conectarse", Toast.LENGTH_SHORT).show();
    }
}
```

El metodo insertarBDUusario insertara los usuarios en la base de datos, en su tabla correspondiente

```
public void insertarBDUsuario ( SQLiteDatabase db, Context context, Usuarios usuario){
    if(db!=null){
        db.execSQL("INSERT INTO Usuarios (nombre, password)" +
        "VALUES ('" +usuario.getNombre() + "', '"+usuario.getPassword()+"')");
    db.close();
        Log.d("prueba2", "todo Bien");
    }
    else{
```

```
            Toast.makeText(context, "Error al conectarse", Toast.LENGTH_SHORT).show();
        }
}
```

El metodo listar devolverá un array de los pedidos insertados en la base de datos

```
public Dest[] listar(SQLiteDatabase db){
        Dest destinoAux;
        Cursor cursor= db.rawQuery(cadListar, null);
        cursor.moveToFirst();
        Dest[] array=new Dest[cursor.getCount()];
        int contador=0;
        do {
                destinoAux=new
Dest(cursor.getString(1),cursor.getString(2),cursor.getInt(3),cursor.getString(4));
                array[contador]=destinoAux;
                contador++;
        }while(cursor.moveToNext());
        cursor.close();
        db.close();
        return array;
}
```

El metodo listarUsuarios, devolvera un array de Usuarios con todos los objetos de tipo usuario

```
public Usuarios[] listarUsuarios(SQLiteDatabase db){
        Usuarios usuarioAux;
        Cursor cursor=db.rawQuery(cadListarUsuarios, null);
        cursor.moveToFirst();
        Usuarios[] array = new Usuarios[cursor.getCount()];
        int contador = 0;
        do{
                usuarioAux=new Usuarios(cursor.getString(1),cursor.getString(2));
                array[contador]=usuarioAux;
                contador++;
        }while(cursor.moveToNext());
        cursor.close();
        db.close();
        return array;
}
```

Por ultimo el metodo onCreate, ejecutara las sentencias sql adecuadas cuando se llegue a esta actividad

```
@Override
public void onCreate(SQLiteDatabase db) {
        db.execSQL(cadSQL);
        db.execSQL(cadSQLUsuarios);
}
```