



Manual Desmontaje del Ordenador

Manual sobre la aplicación

Cristian García Chamizo

ÍNDICE

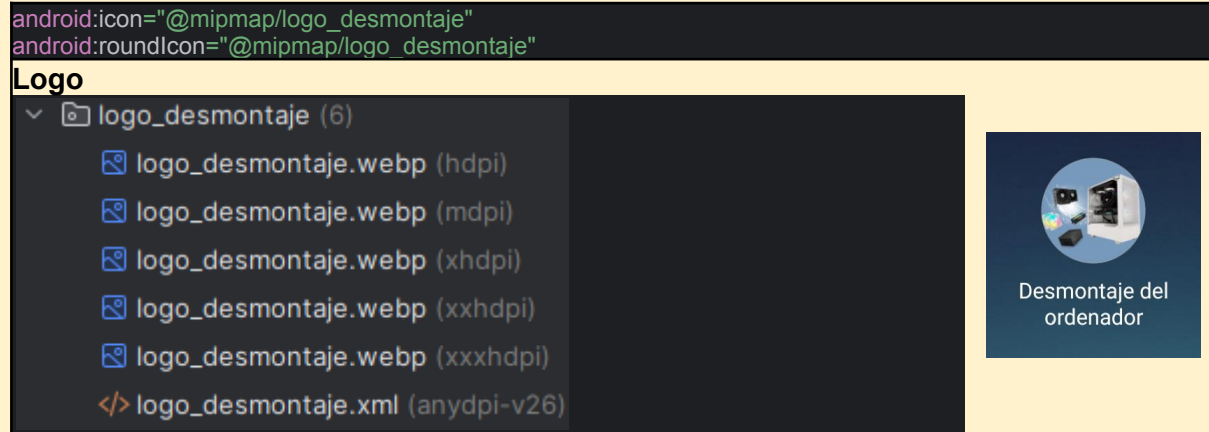
1. Antes de abrir	2
2. Sobre la aplicación	2
3. Portada	3
4. Inicio de sesión	3
5. Pantalla Principal	4
6. Pantalla Contenido	6
7. Menú de opciones	7
a. Valoración	7
b. Descargar el proyecto	7
c. Más información	8
i. Sobre mí	8
ii. Sobre la app	8



1. Antes de abrir

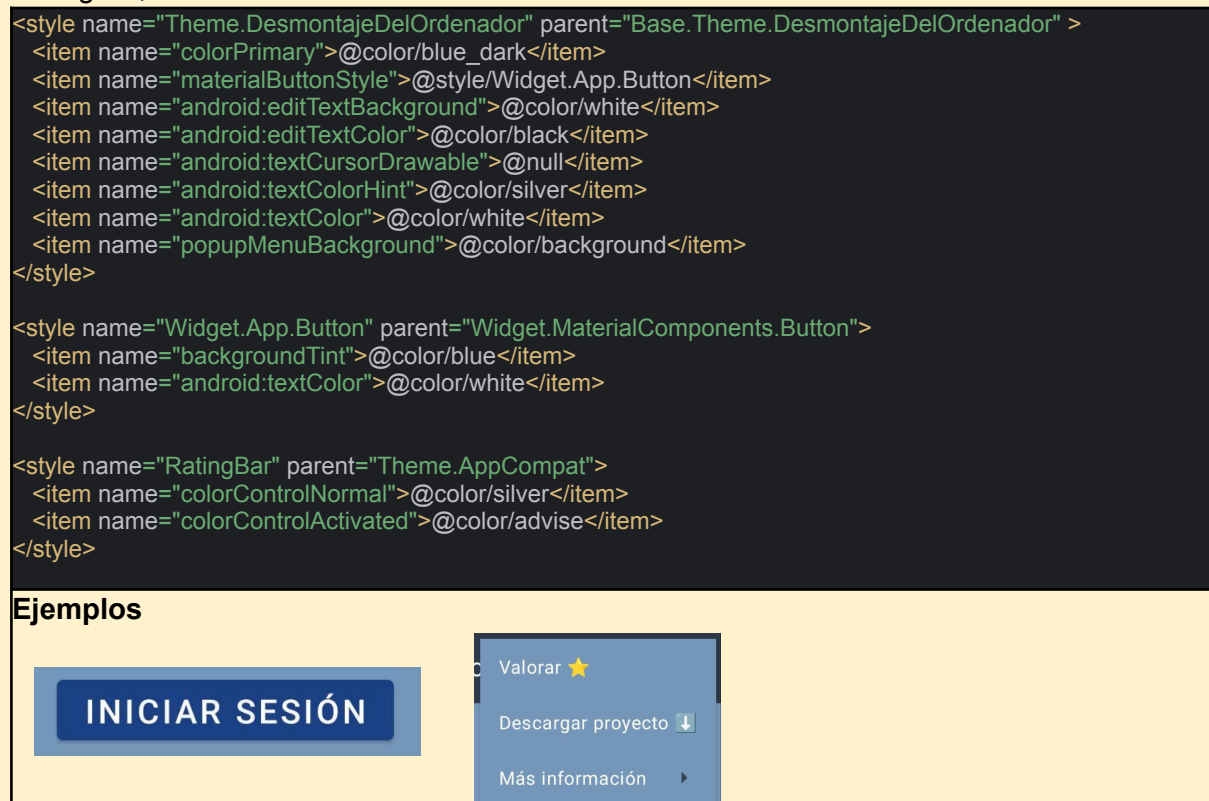
La aplicación “**Desmontaje del ordenador**” tiene un **icono personalizado**.

El icono se ha creado un **Image Asset** en el Resource Manager y asignado en el **Manifest**.



2. Sobre la aplicación

Esta aplicación utiliza un **Theme personalizado**, este establece los **colores predeterminados** del texto, el color de fondo de los botones y menús, el estilo de RatingBar, entre otros.



Los **colores** están almacenados en el archivo **colors.xml**.



3.Portada

Al abrir la aplicación aparece una **pantalla de carga** que utiliza un **ConstraintLayout**, con el título de la aplicación y un **ProgressBar**.

Tras 3 segundos pasa a la siguiente pantalla con un **Intent explícito**.

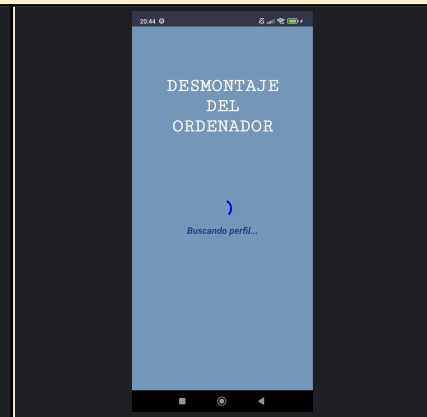
Para esto se crea un **Handler** y se crea un **PostDelayed** de 3s.

```
Handler handler = new Handler();
Intent intent = new Intent(Portada.this, Sesion.class);

final ProgressBar barra = findViewById(R.id.progress);
barra.getIndeterminateDrawable().setColorFilter(Color.BLUE,
PorterDuff.Mode.SRC_IN);

//Buscar si existe un usuario (Proximamente)

handler.postDelayed(new Runnable() {
    @Override
    public void run() {
        startActivity(intent);
    }
}, 3000);
```

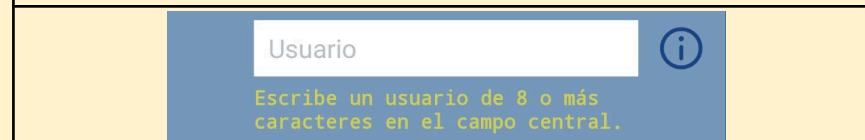


4.Inicio de sesión

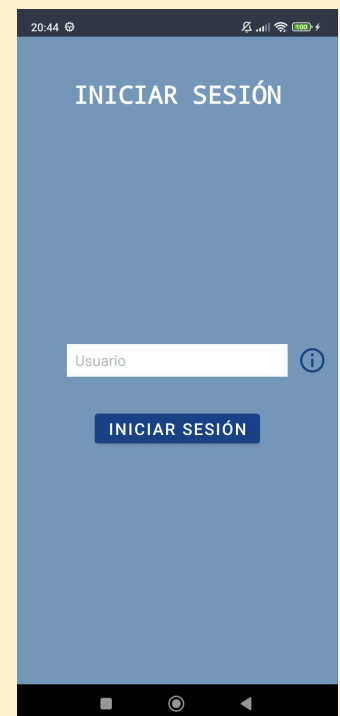
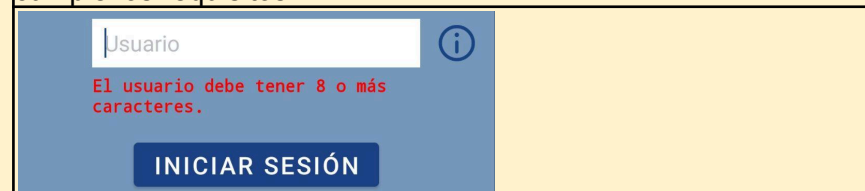
Tras pasar por la portada, carga una actividad con un inicio de sesión sencillo.

Esta pantalla utiliza un **RelativeLayout** y posee un **EditText**, una **imagen** y un **botón**.

Al hacer click en la **imagen**, aparece un aviso sobre los requisitos mínimos.



Al hacer click en el **botón**, aparece un mensaje en rojo si no se cumple los requisitos.



En el **EditText** he añadido un **cursor personalizado** creando un "shape" y asignándole al EditText.

cursor_drawable.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <size android:width="1.5dp"/>
    <solid android:color="@color/blue"/>
</shape>
```

Imagen

```
Usuario
<EditText
...
android:textCursorDrawable="@drawable/cursor_
drawable"/>
```

Al cumplir los requisitos y hacer **click** en el **botón** se nos dirigirá a la **Pantalla Principal**.



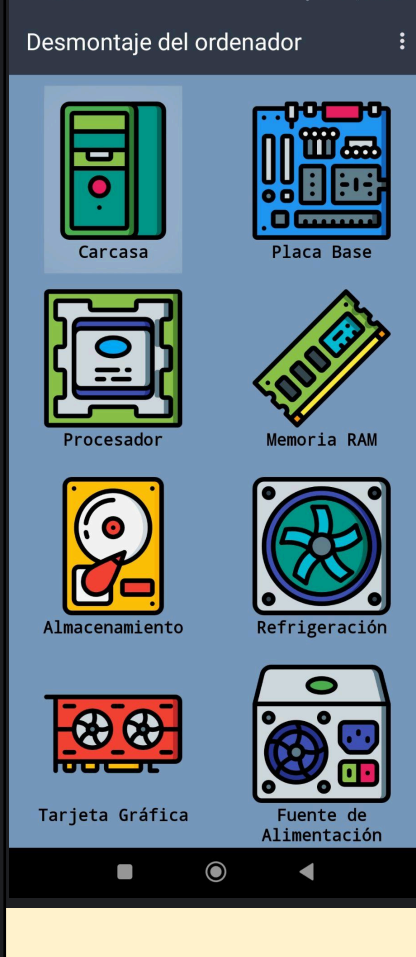
5. Pantalla Principal

La pantalla principal utiliza un **LinearLayout** y se compone de:

- Un **Toolbar** con el **nombre de la aplicación** y un **Menú de opciones** creado en un **Resource File** compuesto de 3 opciones, la tercera opción tiene un **submenú** de 2 elementos.

Menú (Java) <pre> @Override public boolean onCreateOptionsMenu(Menu menu) { getLayoutInflater().inflate(R.menu.menu_principal, menu); return true; } @Override public boolean onOptionsItemSelected(@NonNull MenuItem item) { int opcion = item.getItemId(); if (opcion == R.id.Valorar){ Intent intent = new Intent(Inicio.this, Valoracion.class); startActivity(intent); return true; } else if(opcion == R.id.Info_Autor){ Intent intent = new Intent(Inicio.this, Informacion.class); intent.putExtra("Elemento","Autor"); startActivity(intent); return true; } else if(opcion == R.id.Info_App){ Intent intent = new Intent(Inicio.this, Informacion.class); intent.putExtra("Elemento","App"); startActivity(intent); return true; } else if(opcion == R.id.Descarga){ Intent intent = new Intent(Intent.ACTION_VIEW); intent.setData(Uri.parse("https://github.com/CrisRaptor/Prog.Android-Android_Studio/tree/main/Desmontajedelordenador")); startActivity(intent); return true; } return super.onOptionsItemSelected(item); } </pre>	Menú (Resource File) <pre> <menu xmlns:android="http://schemas.android.com/apk/res/android"> <item android:id="@+id/Valorar" android:title="@string/valorar_menu"/> <item android:id="@+id/Descarga" android:title="@string/descargar"/> <item android:id="@+id/Info" android:title="@string/mas_info"> <menu> <item android:id="@+id/Info_Autor" android:title="@string/sobre_mi"/> <item android:id="@+id/Info_App" android:title="@string/sobre_la_aplicacion"/> </menu> </item> </menu> </pre>
Imágenes 	Toolbar 

- Un **GridView** que utiliza un **adaptador personalizado** compuesto por **una imagen y un texto**. Los elementos **se cargan de un array-string** en el fichero strings.xml. Cada **elemento** tiene un **evento** que envía a otra pantalla (junto con un **extra**).

Adaptador	Clase
<pre> public class Adapt extends BaseAdapter { private ArrayList<Data> data; private Context context; public Adapt(ArrayList<Data> data, Context context) { super(); this.data = data; this.context = context; } @Override public View getView(int position, View view, ViewGroup parent) { View element = LayoutInflater.from(context).inflate(R.layout.element, parent, false); final ImageView imagen = element.findViewById(R.id.imagen); imagen.setImageResource(data.get(position).getImagen()); final TextView nombre = element.findViewById(R.id.nombre); nombre.setText(data.get(position).getNombre()); return element; } @Override public int getCount() {return data.size();} @Override public Object getItem(int position) {return data.get(position);} @Override public long getItemId(int position) {return position;} } </pre>	<pre> public class Data { private int imagen; private String nombre; public Data(int imagen, String nombre) { this.imagen = imagen; this.nombre = nombre; } public int getImagen() { return imagen; } public String getNombre() { return nombre; } } </pre>
Instanciar datos y establecer adaptador	
<pre> //Instancia el listado final GridView list = findViewById(R.id.grid); //Crea la fuente de datos String[] elementos = getResources().getStringArray(R.array.listaComponentes); ArrayList<Data> datos = new ArrayList<>(); for (String elemento : elementos) { String[] partes = elemento.split(","); String nombrelimagen = partes[0]; String texto = partes[1]; int idImagen = getResources().getIdentifier(nombrelimagen, "drawable", getPackageName()); datos.add(new Data(idImagen, texto)); } //Crea el adaptador Adapt miAdapter = new Adapt(datos, this); list.setAdapter(miAdapter); //Crea listener list.setOnItemClickListener(new AdapterView.OnItemClickListener() { @Override public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) { if (adapterView != null) { String selected = ((Data) adapterView.getItemAtPosition(position)).getNombre(); Intent intent = new Intent(Inicio.this, Informacion.class); intent.putExtra("Elemento",selected); startActivity(intent); } } }); </pre>	



6. Pantalla Contenido

Esta pantalla utiliza un **LinearLayout**, posee una estructura fija, pero su contenido cambia según el “extra” recibido por el **Intent explícito**.

Los **datos están almacenados** en el fichero **strings.xml**.

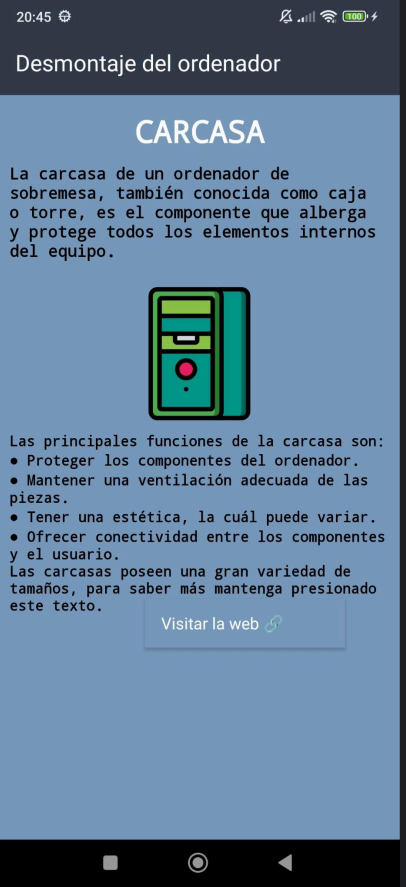
Este layout tiene un **TextView** “título”, un **TextView** “introduccion”, un **ImageView** “imagen”, un **TextView** “descripcion” y un **Button** invisible que sólo se muestra en el submenú “Sobre mí”.

El **TextView** “descripcion” tiene un **Context Menu** que **envía a una web** con contenido relacionado según el “extra” y el contenido utilizando un **Intent implícito**.

Adaptador (switch resumido)

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_contextual, menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    Intent intent = new Intent(Intent.ACTION_VIEW);
    String url;
    switch (elemento){
        case "Carcasa":
            url = "https://www.infootec.net/tipos-caracasas-cajas-ordenador";
            break;
        case "Placa Base": ...
        case "Procesador": ...
        case "Memoria RAM": ...
        case "Almacenamiento": ...
        case "Refrigeración": ...
        case "Tarjeta Gráfica": ...
        case "Fuente de Alimentación": ...
        case "App": ...
        default: ...
            url = "https://github.com/CrisRaptor";
            break;
    }
    intent.setData(Uri.parse(url));
    startActivity(intent);
    return super.onOptionsItemSelected(item);
}
```



Menu (Resource File)

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



7. Menú de opciones

En el **menú de opciones** de la **Toolbar**, se aprecian las siguientes opciones.

a. Valoración

La pantalla utiliza un **RelativeLayout** donde se muestran una serie de apartados compuestos por un **TextView** y un **RatingBar**. Cada **RatingBar** utiliza un **Style** de diseño propio.

Rating bar

```
android:theme="@style/RatingBar"
```

Estilo

```
<style name="RatingBar" parent="Theme.AppCompat">
  <item name="colorControlNormal">@color/silver</item>
  <item name="colorControlActivated">@color/advise</item>
</style>
```

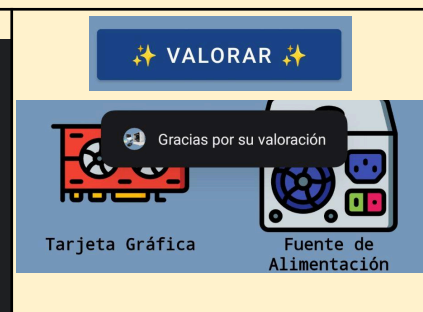


Al final de los apartados hay un **Botón** con un **evento** finalizando esta actividad y muestra un **Toast** "Gracias por su valoración".

Evento del botón

```
Button valorar = findViewById(R.id.btnValorar);

valorar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(Valoracion.this, "Gracias por su valoración",
            Toast.LENGTH_SHORT).show();
        finish();
    }
});
```



b. Descargar el proyecto

Esta opción se trata simplemente de un **Intent implícito** de una **url** al repositorio de la aplicación en github.

Inicio (onOptionsItemSelected)

```
if(opcion == R.id.Descarga){
    Intent intent = new Intent(Intent.ACTION_VIEW);

    intent.setData(Uri.parse("https://github.com/CrisRaptor/Prog.Android-Android_Studio/tree/main/Desmontajedelord
enador"));
    startActivity(intent);
    return true;
}
```




c. Más información

Este menú se compone de **2 submenús** con información del autor y la aplicación.

i. Sobre mí

Utiliza el **layout de contenido**, mostrando además el **Botón** “Contacta conmigo” que tiene un **Intent implícito** para enviar un **correo**, este incluye un receptor, asunto y mensaje inicial.

<pre>Switch case "Autor": contenido = getResources().getStringArray(R.array.autor_contenido); cargarContenido(contenido); Button botonContactar = findViewById(R.id.btnContactar); botonContactar.setVisibility(View.VISIBLE); botonContactar.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent intent = new Intent(Intent.ACTION_SEND); intent.setType("text/plain"); intent.putExtra(Intent.EXTRA_EMAIL, new String[] {"cgarcha0608@g.educaand.es"}); intent.putExtra(Intent.EXTRA_SUBJECT, "Sobre la aplicación..."); intent.putExtra(Intent.EXTRA_TEXT, "Buenas Cristian,"); startActivity(intent); } }); break;</pre>	
--	--

ii. Sobre la app

Este submenú utiliza el **layout de contenido**, con información sobre la aplicación.

FIN