

# XML

## Android Studio

### Interfaze diseinua XML-ean oinarrituta

### PAAG – 2023-2024

# Reto 1. Interfaze diseinua XML-ean oinarrituta

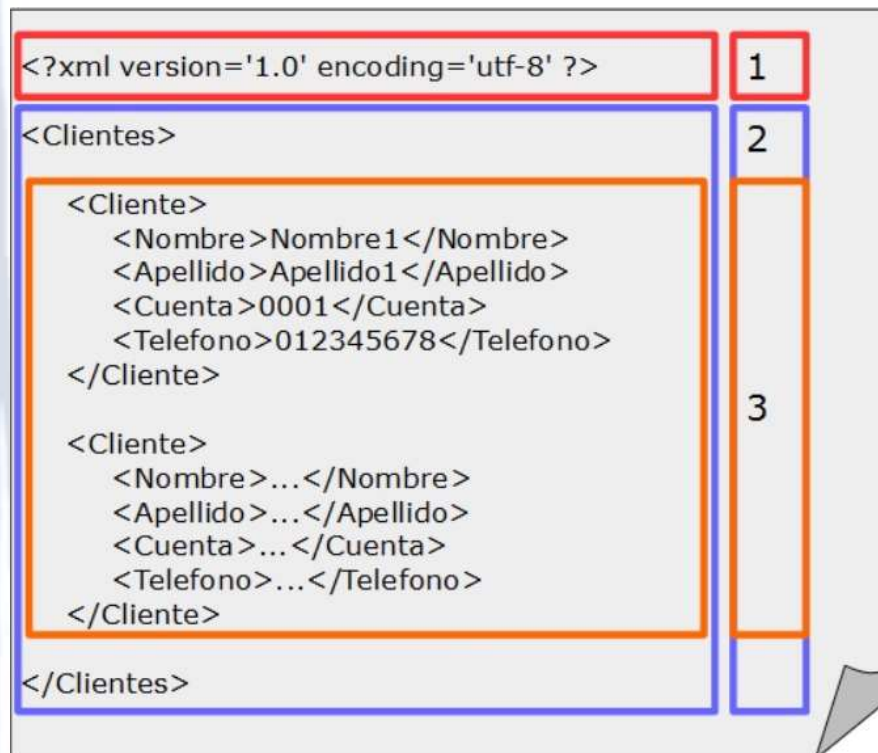
## Indizea

- XML lengoaia
- XML Androiden
  - Oinarrizko kontzeptuak
  - XML fitxategien motak
  - Adibide bat
  - Adibide bi
  - Figma erabiltzen
- Figma

# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML lengoaia

- XML-a markatze lengoaia da. XML-an idatzitako edozein gauzak bi zati ditu: datuak eta etiketak. Etiketa hauek datuen egitura eta aurkezpenaren informazioa ematen digute.
- Estandar ireki bat da. Edozein informazio mota, testuinguru jakin baten egokitutako formatu batean biltegitatu eta antolatzeko aukera ematen du.
- Informazioa trukatzeko pentsatuta dago. Edozein plataforma eta jatorritatik independentea, eta, adibidez, sistema eragile desberdinetan implementatutako datu-base desberdinen artean erabil daiteke.



- (1) hitzaurrea (aukerakoa), XML dokumentu bat dela eta erabilitako kodifikazio mota zehazten duena
- (2) Erroko elementu edo nodoa (nahitaezkoa), dokumentuaren gorputza osatuko duena eta aitortutako gainerako elementu semeak bilduko dituena.
- (3) Semeak edo erroko elementuaren adar-nodoak; horiek, aldi berean, ume-nodoz osatuta egongo dira, eta horietan testua ezarriko da.

# Reto 1. Interfaze diseinua XML-ean oinarrituta

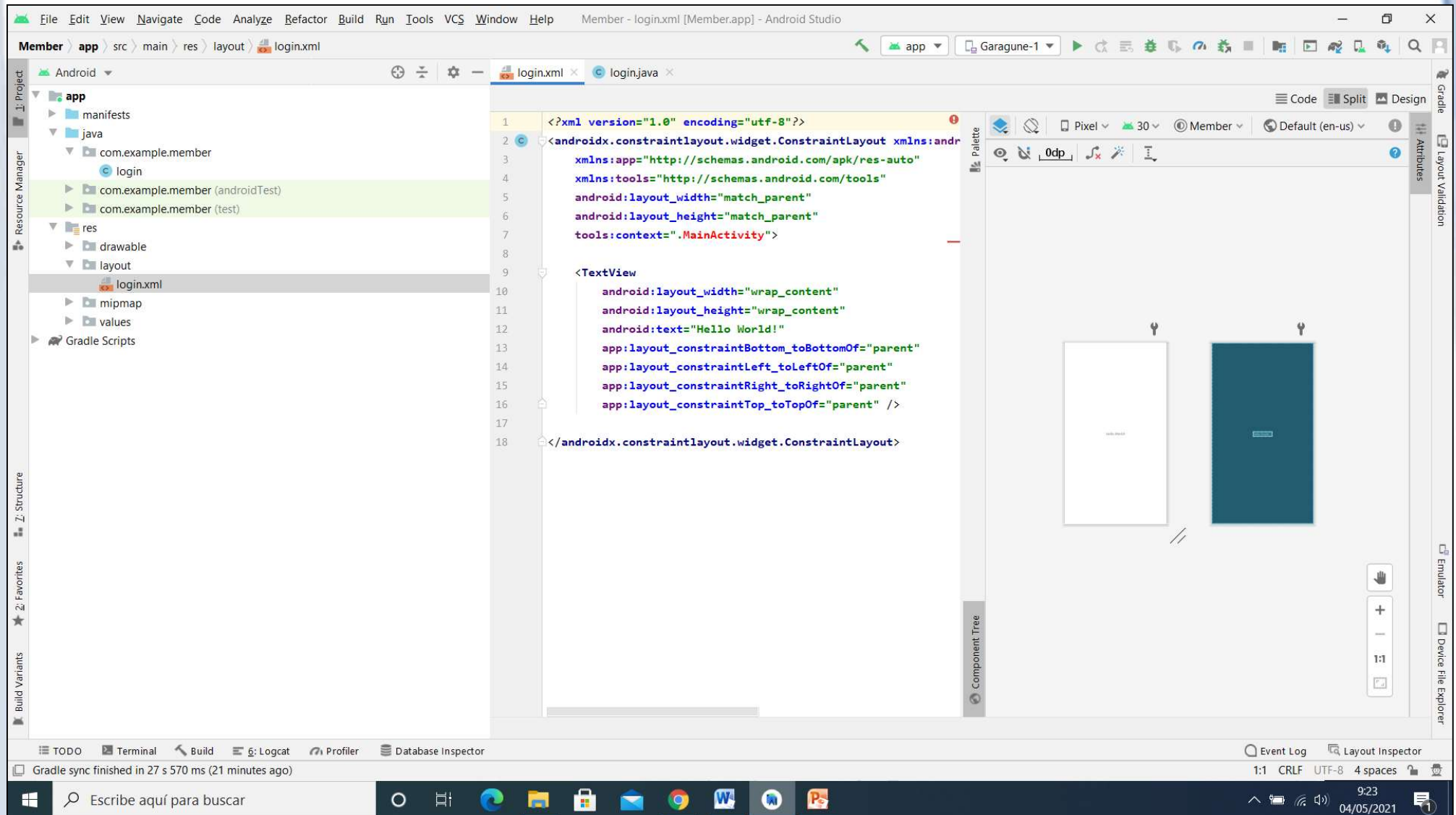
## XML lengoaia

- Android Studio erabiliko dugu mobilarentzako aplikazio bat egiteko. Pantailak / Interfaze grafikoak diseinatu beharko ditugu
- XML-a interfaze grafiko hauek deskribatzeko erabil daiteke.
- Android-ek aukera hori ematen du: XML-n oinarritutako diseinu-fitxategiekin interfazea sortzea eta/edo aldatzea.

Beraz, Android Studio instalatu behar dugu: <https://developer.android.com/>

# Reto 1. Interfaze diseinua XML-ean oinarrituta

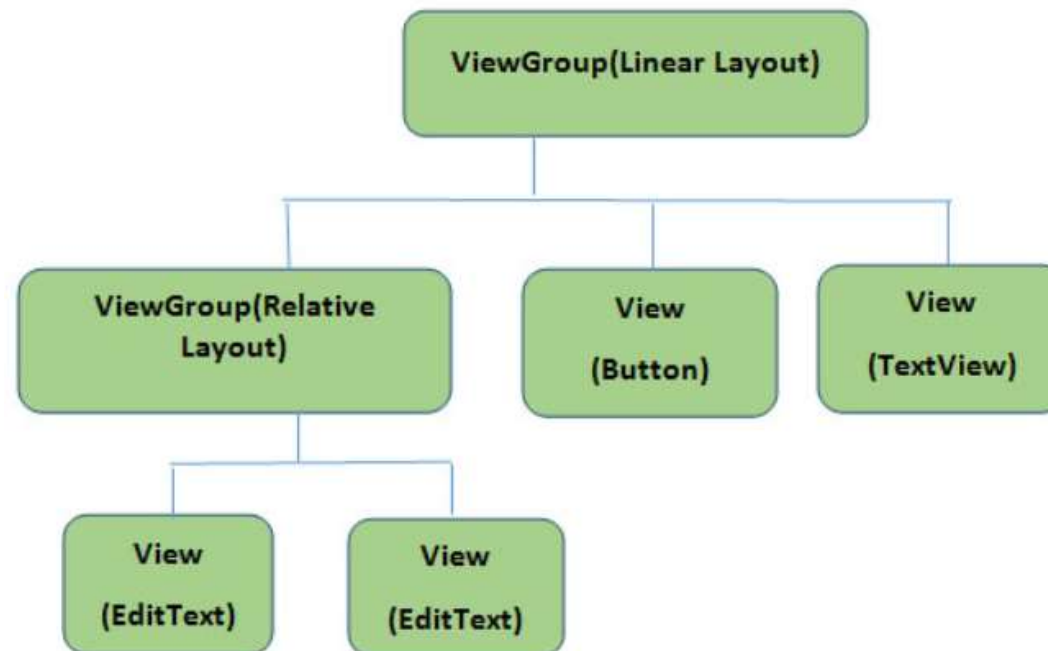
## XML Androiden: erabilitako oinarrizko kontzeptuak eta XML fitxategi motak



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: oinarrizko kontzeptuak

- Androiden erabiltzaile-interfazearen kontzeptu osoa **View** eta **ViewGroup** objektuen hierarkia erabiliz definitzen da.
- ViewGroup bigarren mailako bistak antolatzen dituen edukiontzi ikusezina da.
- Bigarren mailako bista hauek interfazearen zati ezberdinak sortzeko erabiltzen diren beste **widget** batzuk dira.
- ViewGroup batek beste ViewGroup bat izan dezake bigarren mailako elementu gisa

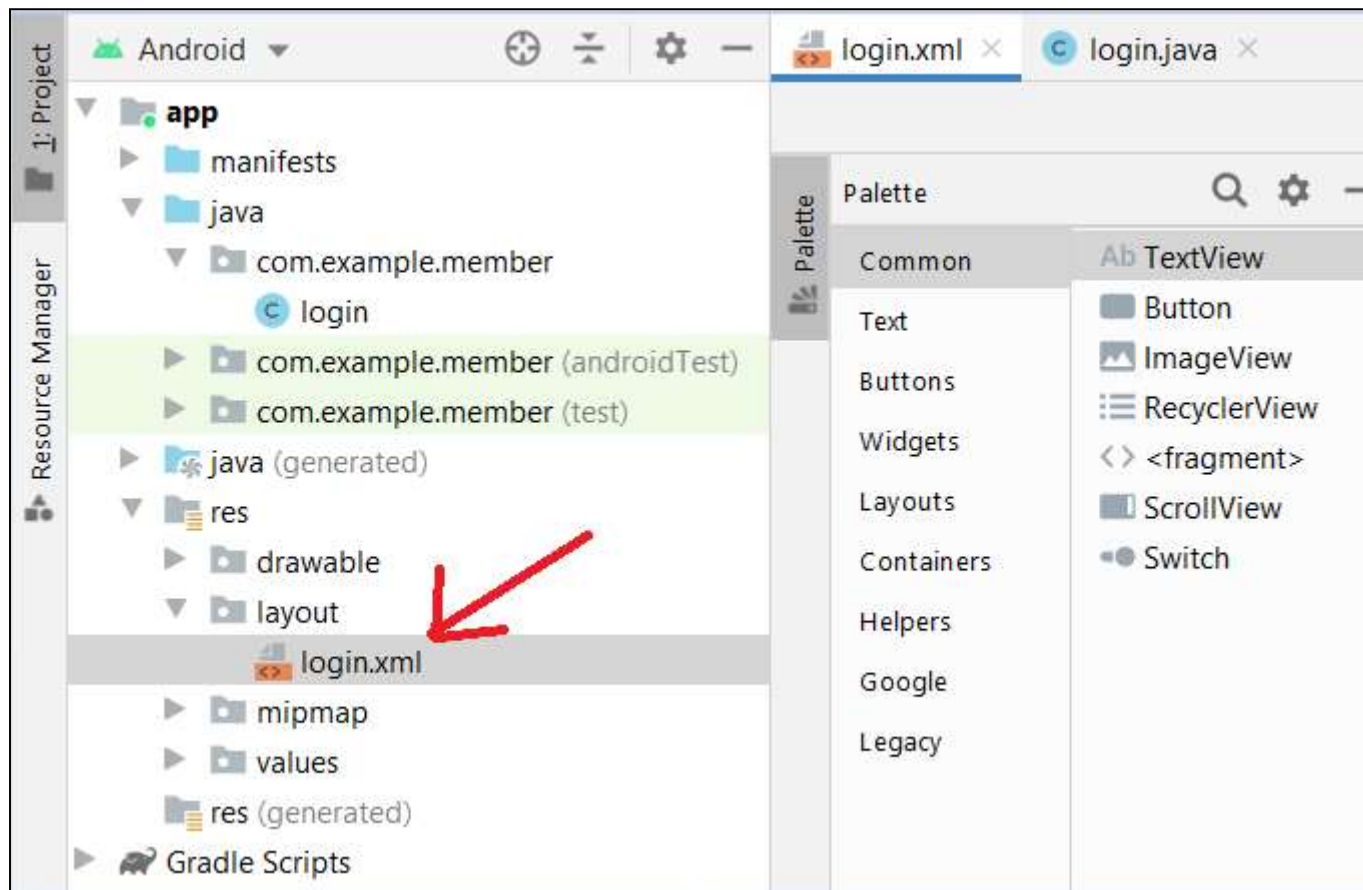




# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: XML fitxategi motak

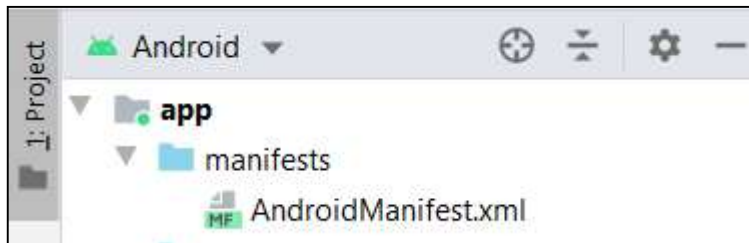
- **Layout.xml:** diseinuko xml fitxategiak gure aplikazioaren benetako erabiltzaile-interfazea definitzeko erabiltzen dira. Aplikazioan erabili nahi ditugun elementu guztiak (bistak) edo tresnak jasotzen ditu. TextView-ak bezala, botoiak eta erabiltzaile-interfazearen beste elementu batzuk.



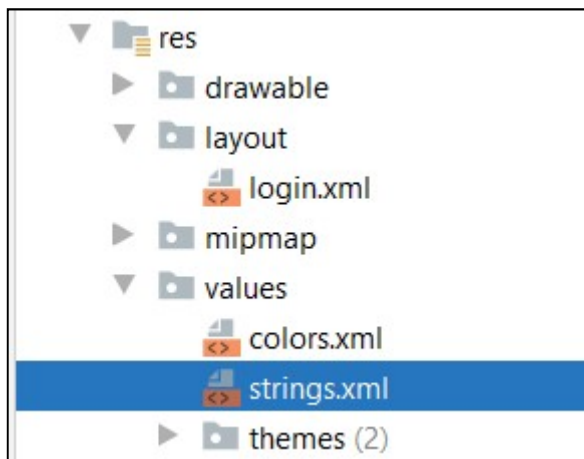
# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: XML fitxategi motak

- **Manifest.xml:** xml fitxategi hau gure aplikazioaren osagai guztiak definitzeko erabiltzen da. Barne hartzen ditu gure aplikazioko paketeen izenak, gure Activity-ak, zerbitzuak eta gure aplikazioak behar dituen baimenak. Adibidez: demagun gure gailuaren kamera erabili behar dugula, orduan, gure Manifestoan kamera erabiltzeko baimena zehaztu behar dugu.



- **Strings.xml:** xml fitxategi hau kate bakar batekin kodifikatutako kateak ordezkatzeko erabiltzen da. Xml fitxategi honetan kate guztiak definituko ditugu, eta, gero, fitxategi horretara sartuko gara gure aplikaziotik. Artxibo horrek aukera ematen digu kodearen berrerabilpena hobetzeko.

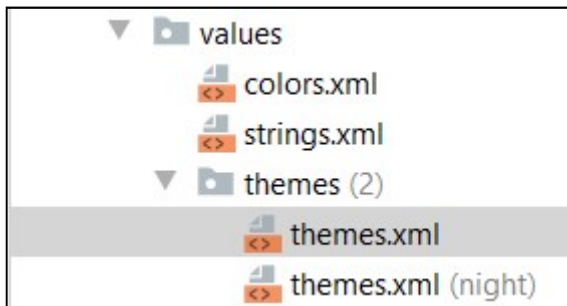




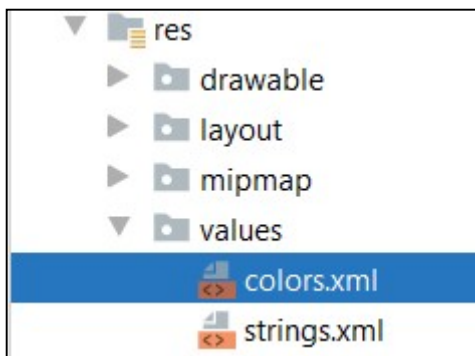
# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: XML fitxategi motak

- **Themes.xml:** xml hau aplikazioaren estilo ezberdinak definitzeko erabiltzen da. Artxibo honetan zehazten ditugu gure gaiak eta estilo pertsonalizatuak.



- **Color.xml:** fitxategi hau gure aplikazioan erabiltzen ditugun kolore-kodeak definitzeko erabiltzen da



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Adibide bat

"IERabiltzaileaXML" deituko dugun proiektua egingo dugu, non layout mailako kontrolak definituko ditugu XML fitxategietan

### **IERabiltzaileaXML Proiektua**

- Aplikazio sinple bat sortu (**Empty Activity**): pantaila bat, botoi bat eta mezu bat.
- Behar ditugun osagaiak:
  - **MainActivity** klasea, Activity klase oinarritik heredatzen duena, eta instantziak sortzeko eta erabiltzaile-interfazearen propietateen diseinu programatikoa egiteko logika definituko duena
  - Layout **activity\_main.xml**, layout batez osatua, exekuzio denboran eraikitzen diren gainerako elementuetarako oinarri gisa balioko duena.

**<androidx.constraintlayout.widget.ConstraintLayout .....>**

"A ConstraintLayout is a [android.view.ViewGroup](#) which allows you to position and size widgets in a flexible way"

Informazio gehiago:

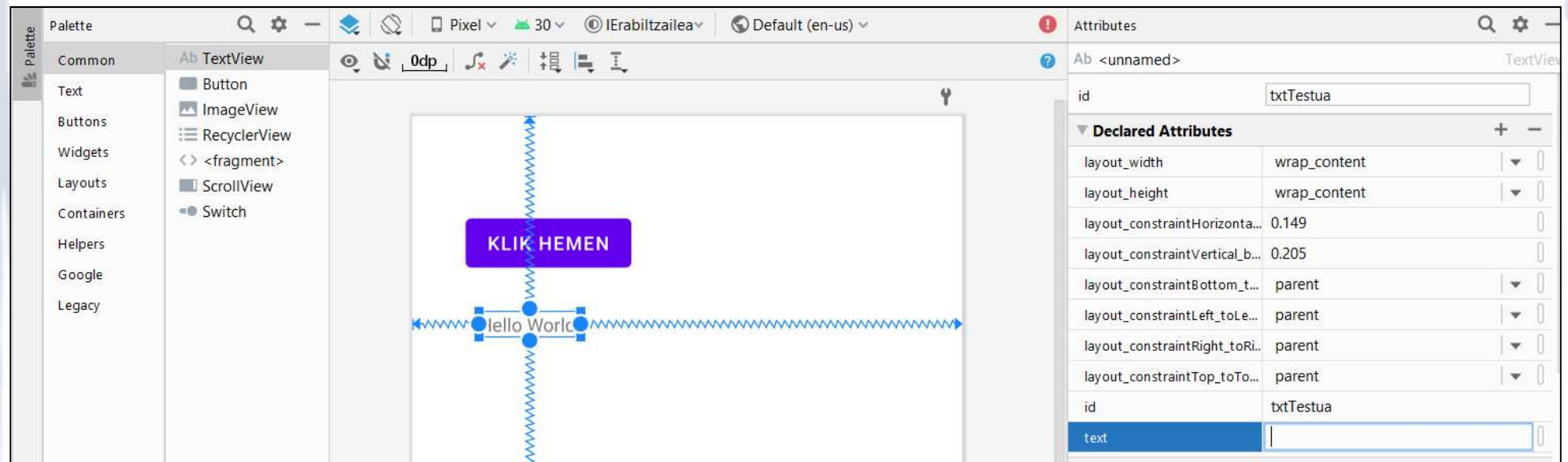
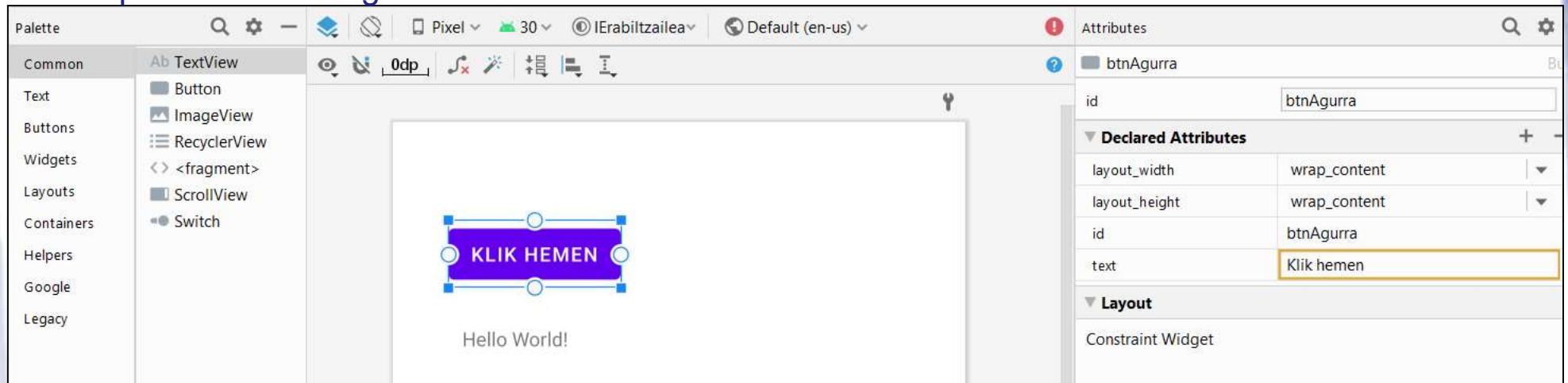
<https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout>

- Nola egingo dugu:
  - **res/layout/activity\_main.xml** fitxategian klik egin eta design pestainean jarriko gara.

# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Adibide bat

- Button aukeratu eta arrastatu eta aurretik eduki dugun TextView («Hello World») aprobe txatuko dugu. Id eta text aldatu behar dira



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Adibide bat

- Orain java fitxategian sortu den kodigoa begiratuko dugu eta behar dena gehituko dugu:  
**MainActivity.java**

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private TextView txtTestua;
    private Button btnAgurra;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        txtTestua = (TextView)findViewById(R.id.txtTestua);
        btnAgurra = (Button)findViewById(R.id.btnAgurra);

        btnAgurra.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                txtTestua.setText("Kaixo Mundua!!!");
            }
        });
    }
}
```

# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Adibide bat

- Orain XML fitxategian sortu den kodigoa begiratuko dugu:

.....

```
<TextView
    android:id="@+id/txtTestua"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.149"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.205" />
```

```
<Button
    android:id="@+id/btnAgurra"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Klik hemen"
    tools:layout_editor_absoluteX="40dp"
    tools:layout_editor_absoluteY="71dp" />
```

.....

- Eta botoia pertsonalizatzen saiatuko gara:

# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Adibide bat

- Baliabide bi sortuko ditugu: res – drawable (eskuineko klik) – new drawable resource file: **botoia.xml** eta **txtbotoia.xml**
- Aldaketa batzuk egingo ditugu gure erara jartzeko:

### **botoia.xml**

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

    <corners android:radius="150dp" />

    <stroke android:color="@android:color/white" android:width="5px" />

</shape>
```

### **txtbotoia.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:color="#ffffff" android:state_pressed="false" />
    <item android:color="#aa023d61" android:state_pressed="true" />
</selector>
```



# Reto 1. Interfaze diseinua XML-ean oinarrituta

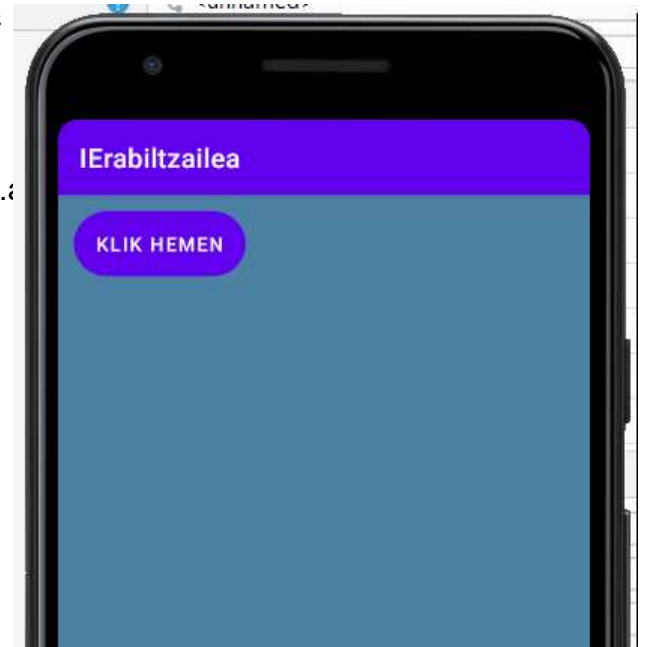
## XML Androiden: Adibide bat

- Fitxategi nagusian aldatuko dugu orain: **activity\_main.xml**

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="#4d81a2"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:orientation="vertical"
    android:padding="12dp"
    tools:context=".MainActivity">
```

```
.....
<Button
    android:id="@+id/btnAgurra"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="15dp"
    android:background="@drawable/botoia"
    android:textColor="@drawable/txt_botoia"
    android:text="Klik hemen"
    tools:layout_editor_absoluteX="40dp"
    tools:layout_editor_absoluteY="71dp" />
```

- Aldaketak ikusi aplikazioa martxan jarrita
- eta botoian klik eginda



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Adibide bat

- Activity berria gehituko dugu (layout karpeta ganean eskuineko botoiarekin «New - Activity – Empty Activity - **Bigarrena**»
  - Layout berrian Checkbox bat gehituko dugu:

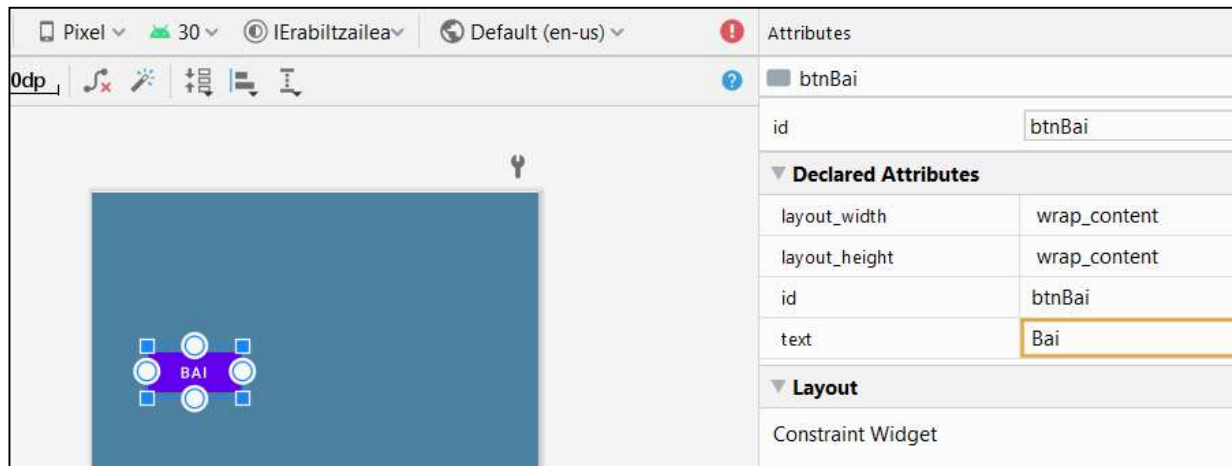
Testua ="English"

```
<CheckBox
  android:id="@+id/checkBox"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="English"
  tools:layout_editor_absoluteX="71dp"
  tools:layout_editor_absoluteY="230dp" />
```

# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Adibide bat

- Botoi bat gehitu («BAI»):



- Bigarren botoi bat gehitu («EZ» - Saiatu XML-a kopiauta eta behar diren aldaketak egiten) :

```
<Button  
  android:id="@+id/btnBai"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Bai"  
  tools:layout_editor_absoluteX="26dp"  
  tools:layout_editor_absoluteY="130dp" />
```

```
<Button  
  android:id="@+id/btnEz"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Ez"  
  tools:layout_editor_absoluteX="145dp"  
  tools:layout_editor_absoluteY="130dp" />
```

# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Adibide bat

- **Strings.xml** erabiliko dugu testu katai batzuk prestatzeko (res – values – strings.xml)

```
<resources>
  <string name="app_name">IErabilitzailea</string>
  <string name="bai_eu">Bai</string>
  <string name="ez_eu">Ez</string>
  <string name="bai_en">Yes</string>
  <string name="ez_en">No</string>
</resources>
```

- **Bigarrena.java**

```
checkBox = (CheckBox) findViewById(R.id.checkBox);
btnBai = (Button) findViewById(R.id.btnBai);
btnEz = (Button) findViewById(R.id.btnEz);

checkBox.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v)
    {
        if (checkBox.isChecked())
        {
            btnBai.setText(R.string.bai_en);
            btnEz.setText(R.string.ez_en);
        }
        else
        {
            btnBai.setText(R.string.bai_eu);
            btnEz.setText(R.string.ez_eu);
        }
    }
});
```

# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Adibide bat

- **AndroidManifest.xml**

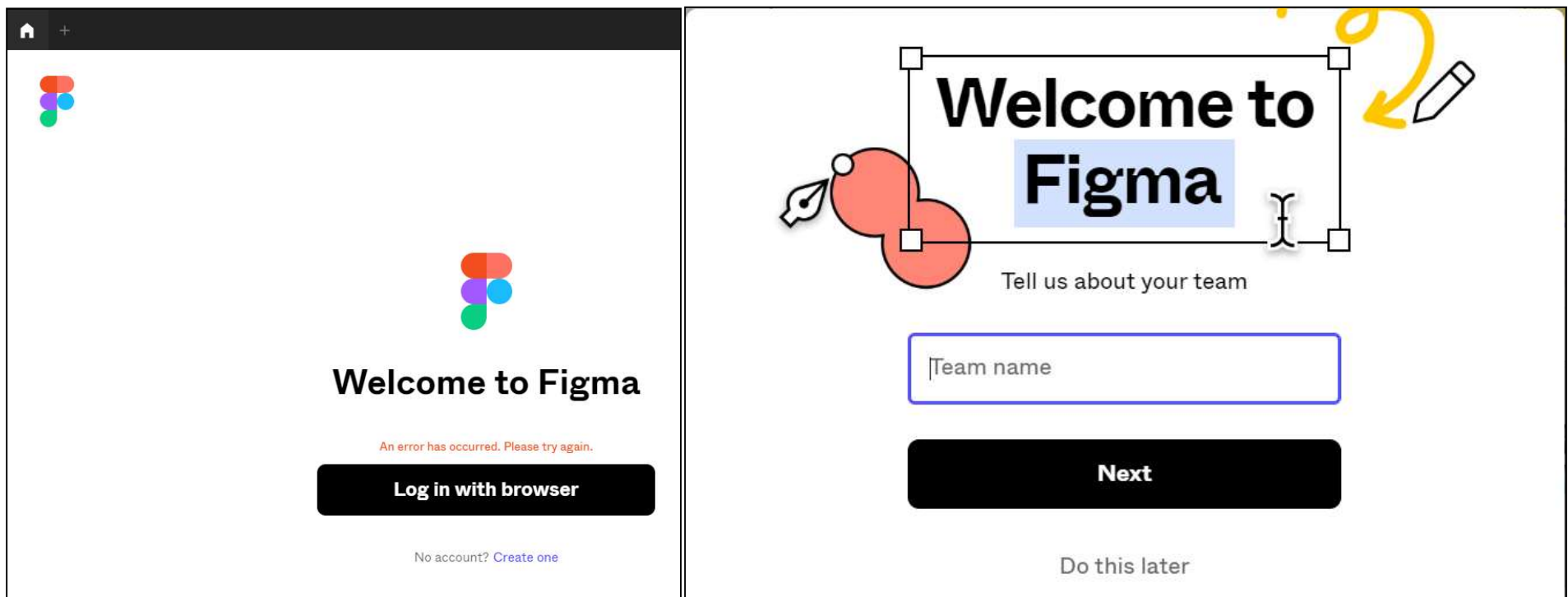
```
<activity android:name=".Bigarrena">  
  <intent-filter>  
    <action android:name="android.intent.action.BIGARRENA" />  
  
    <category android:name="android.intent.category.LAUNCHER" />  
  </intent-filter>  
</activity>
```

# Reto 1. Interfaze diseinua XML-ean oinarrituta

## Figma

Proiektu bat diseinatzeko beharrezkoak diren tresna guztiak eskaintzen dituen programa da Figma. Prototipoak sortzeko, bererabiltzeko kodea sortzeko (hand-off) eta ilustratzeko aukera ematen du.

Instalatuta eta gero saioa hasi nabigatzailea erabilita «Log in with browser» eta kontua sortu Gmaileko datuekin. Taldea sortu daiteke kolaboratzaileekin. «Design with Figma» aukeratuta.

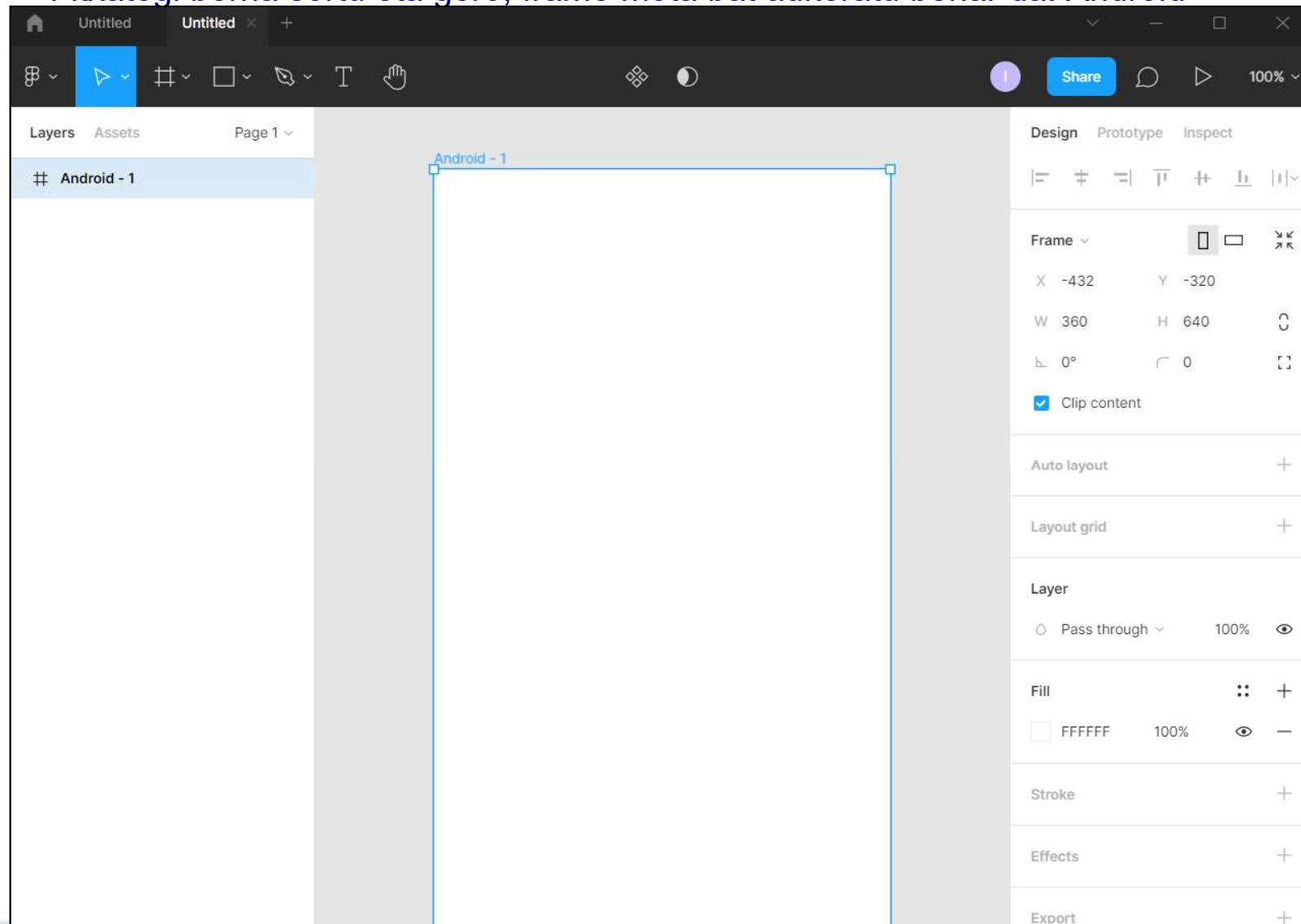




# Reto 1. Interfaze diseinua XML-ean oinarrituta

## Figma

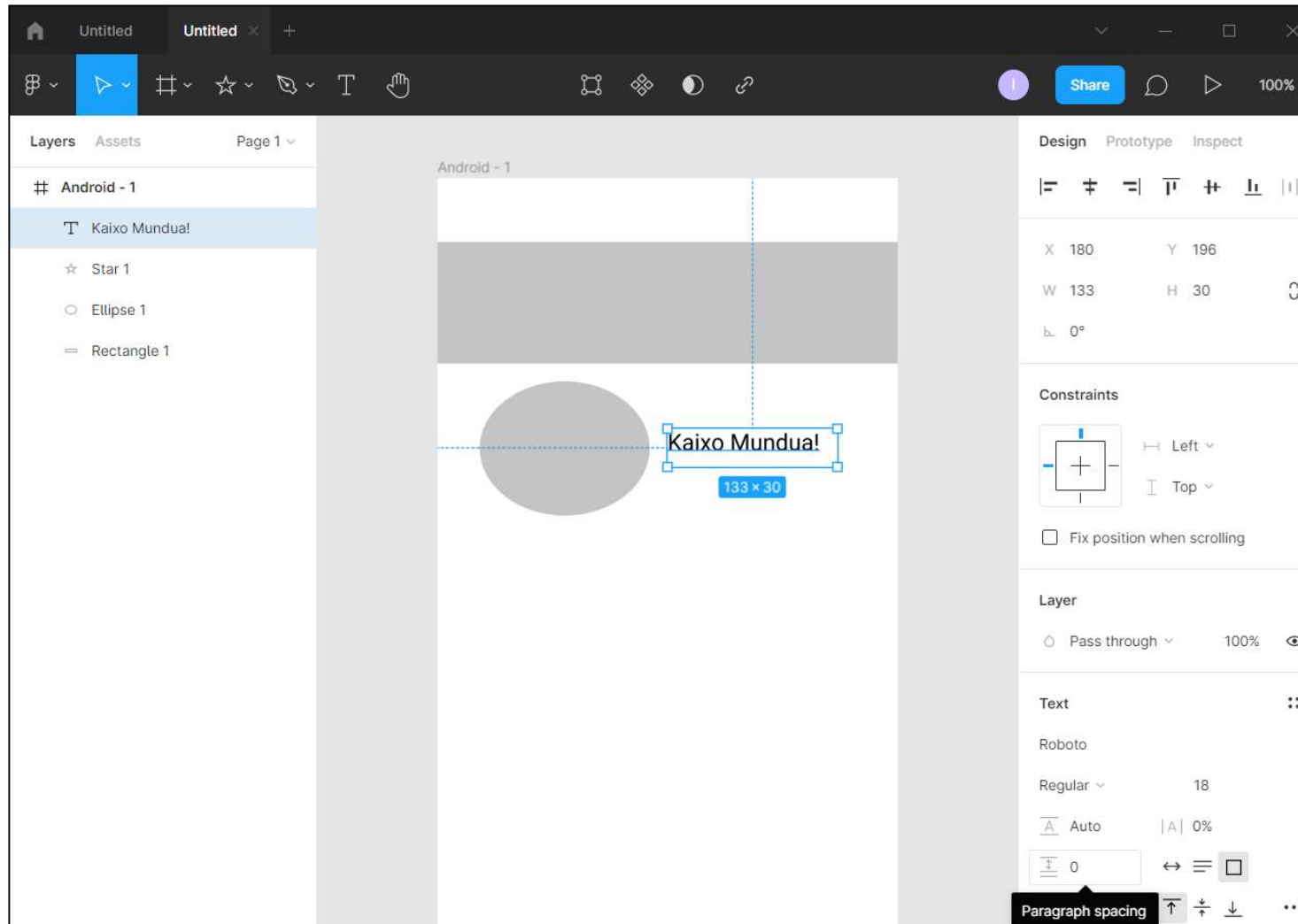
Fitxategi berria sortu eta gero, frame mota bat aukeratu behar da: Android



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## Figma

Osagai ezberdinak gehitu daiteke goiko menu beltzetik eta gero propietateak aldatu eskuineko aukerak erabiliz

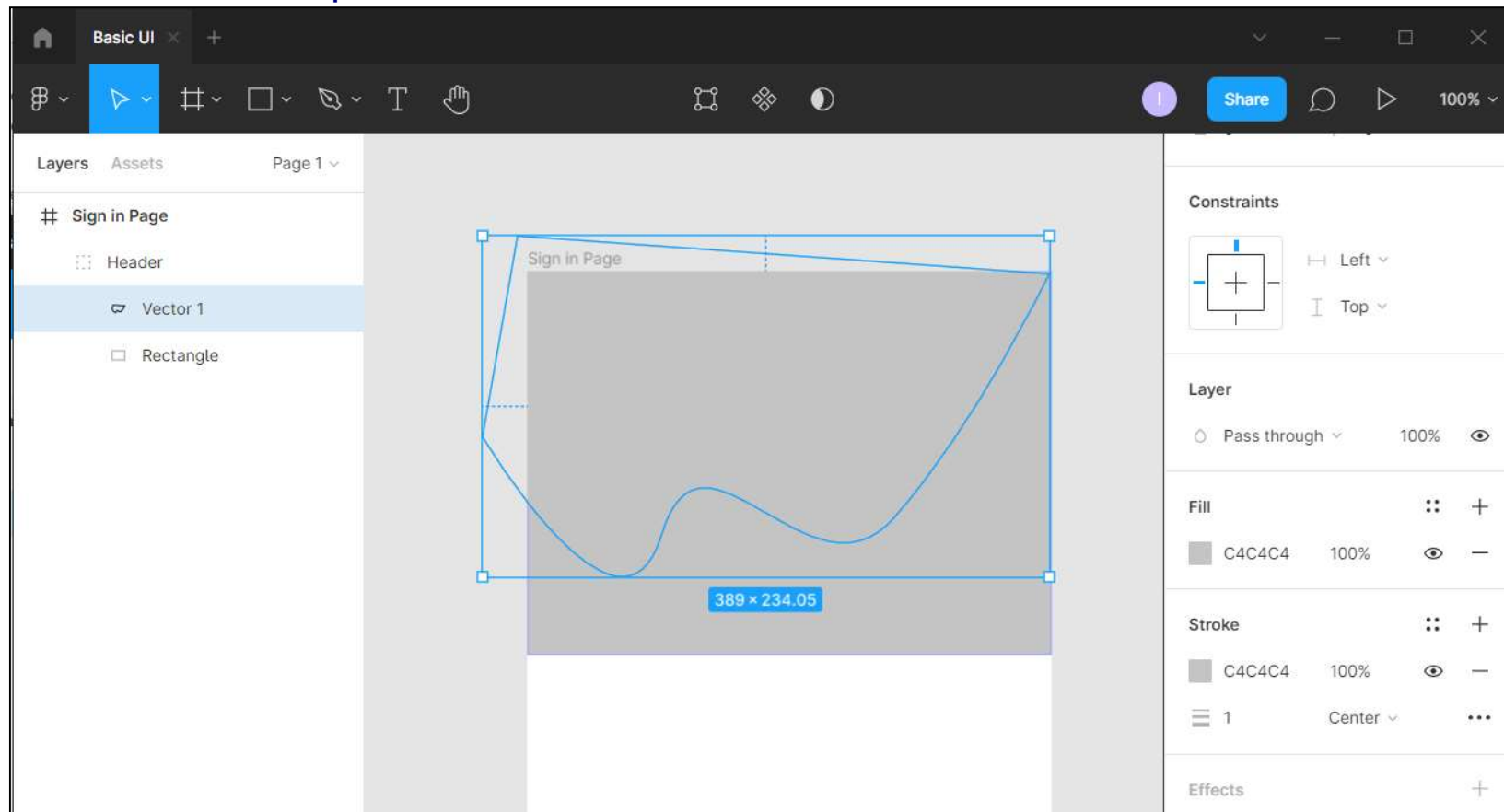


# Reto 1. Interfaze diseinua XML-ean oinarrituta

## Figma

Zerotik hasita berriro, «sign in» egiteko pantaila diseinatuko dugu: **Basic UI** izenarekin

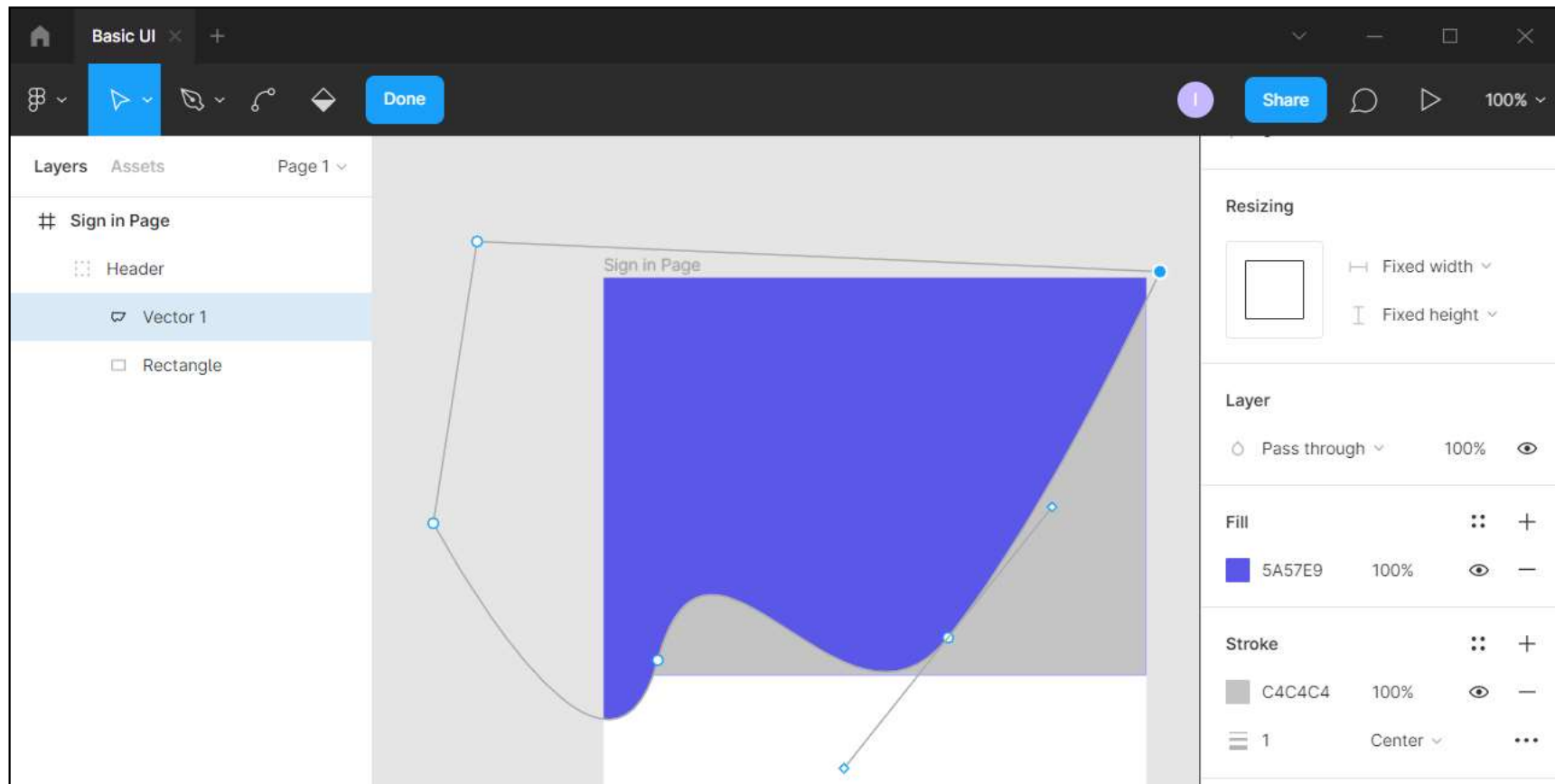
- Frame berria prestatu (Android) eta izena jarri: **Sign in Page**
- Talde bat sortu, **Header**, eta barruan Laukizuzena eta gainetik idazluma eta angulu ezberdinakin probatuta.



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## Figma

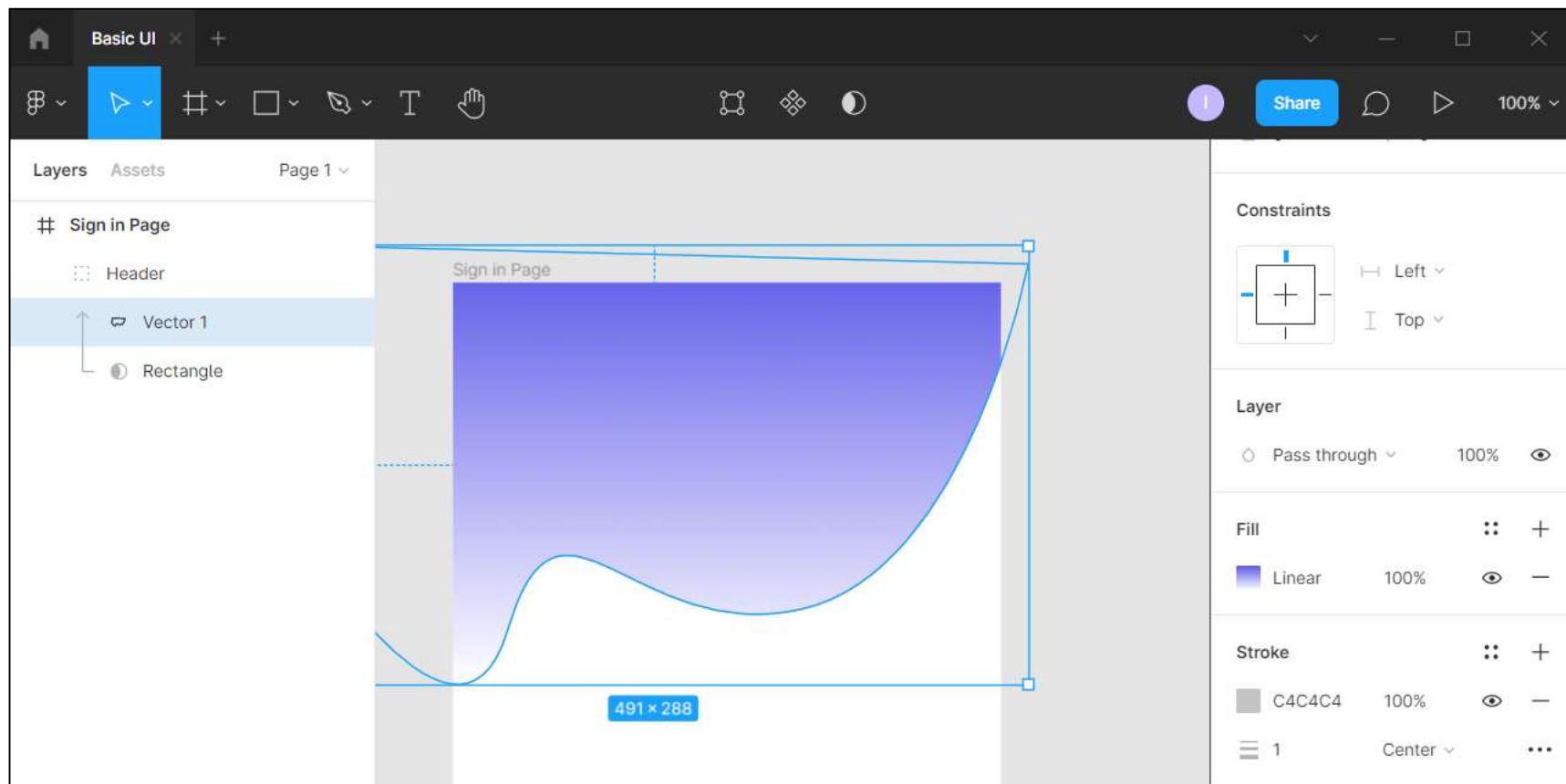
- Bektorea eskalatu nahi den diseinua lortu arte
- Atzealdeko kolorea ere aldatu



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## Figma

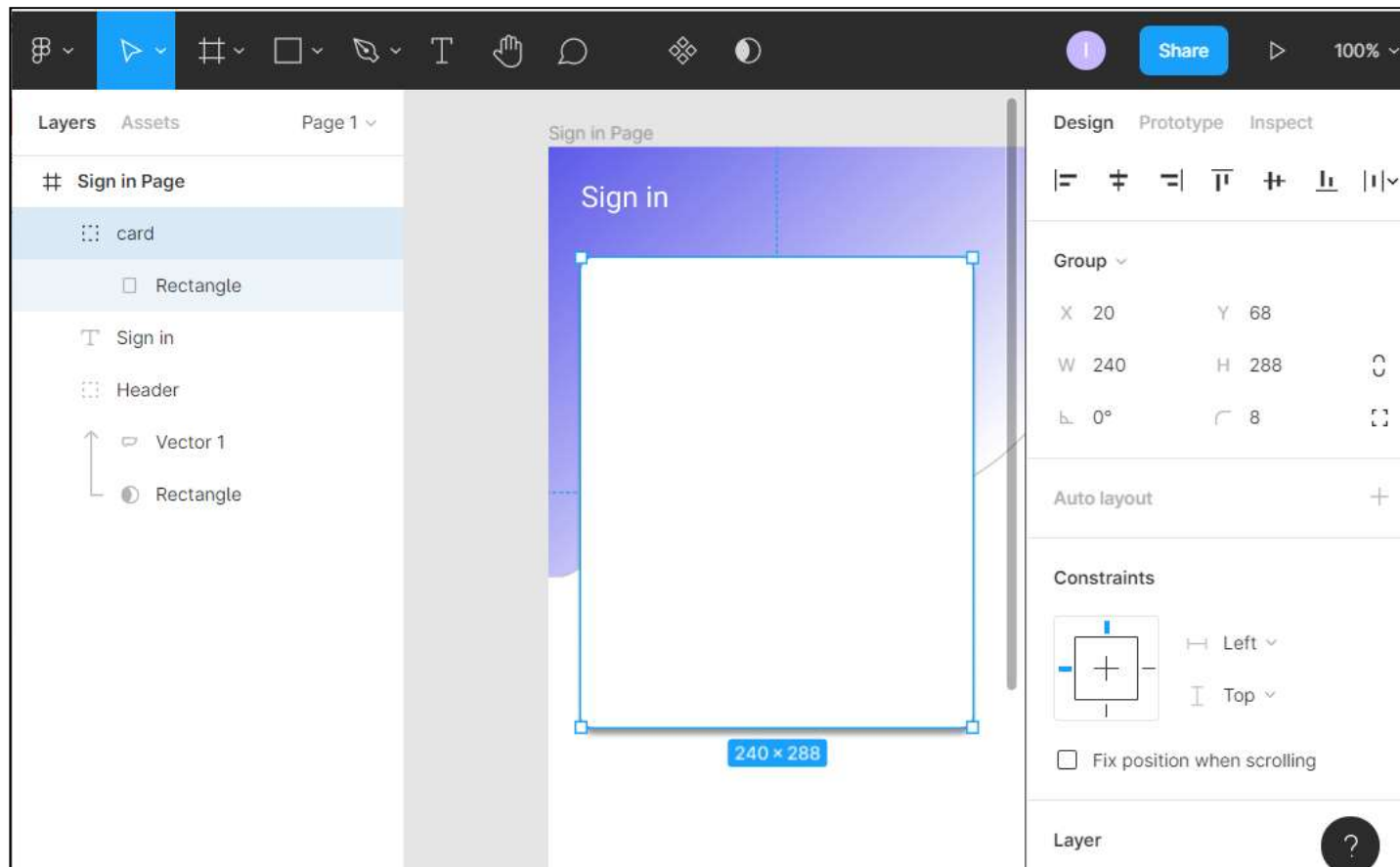
- Laukizuzena aukeratu eta «use as mask» eta ajusteak egin
- Kolorea ere «Linear» aukeratu eta gradiente bezala erakutsi



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## Figma

- Testua gehitu «Sign in» eta propietateetan pertsonalizatu (X 20, Y 20, letra zuria)
- Laukizuzena marraztu eta «card» izena jarri (eskuineko botoia eta group selection aukeratu) (X 20, atzealdea zuria, corner radius 8)

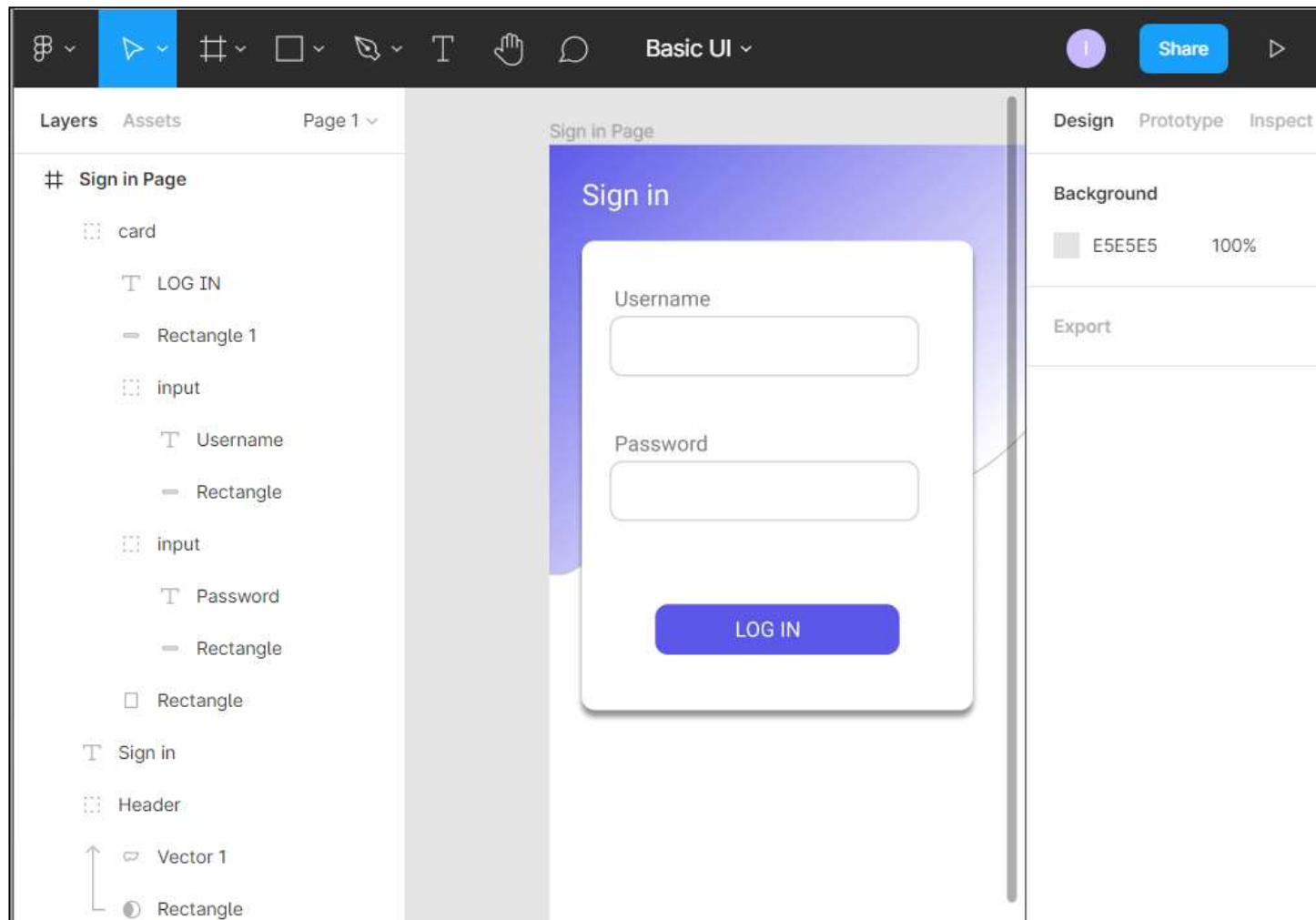




# Reto 1. Interfaze diseinua XML-ean oinarrituta

## Figma

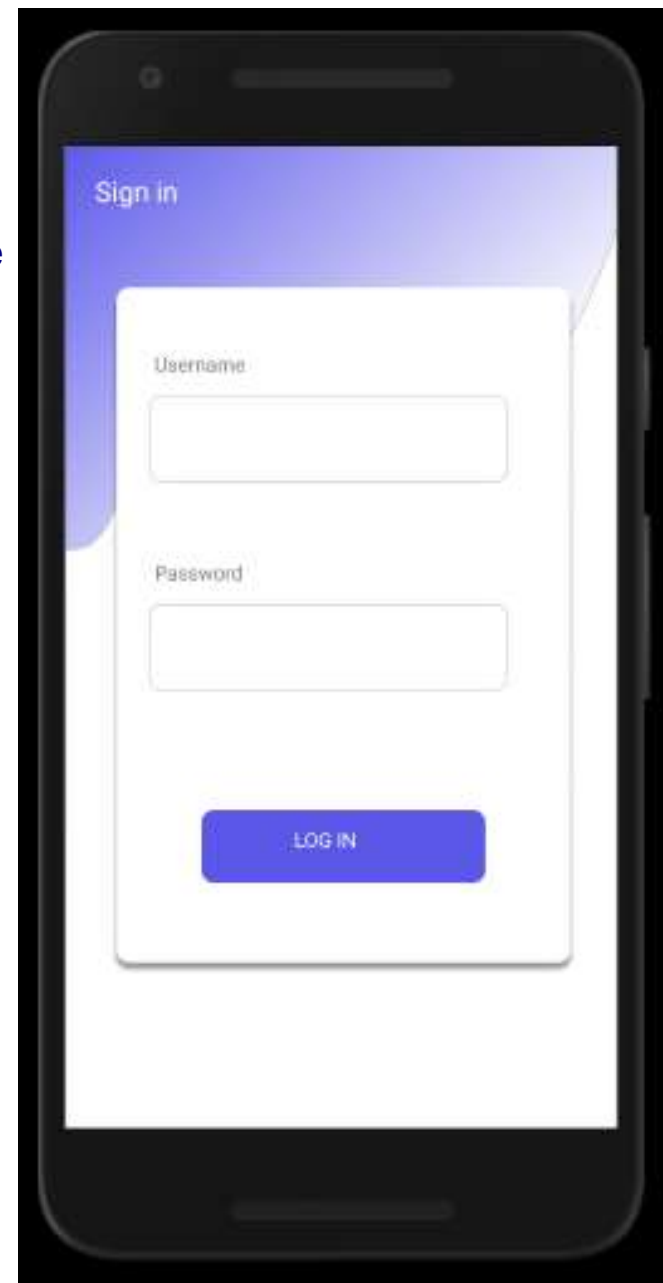
- Username eta Password sartzeko laukizuzenak eta testuak jarriko ditugu, bakoitza talde batean
- Gainera beste lauki zuena eta testua botoia egiteko



## Reto 1. Interfaze diseinua XML-ean oinarrituta

Figma

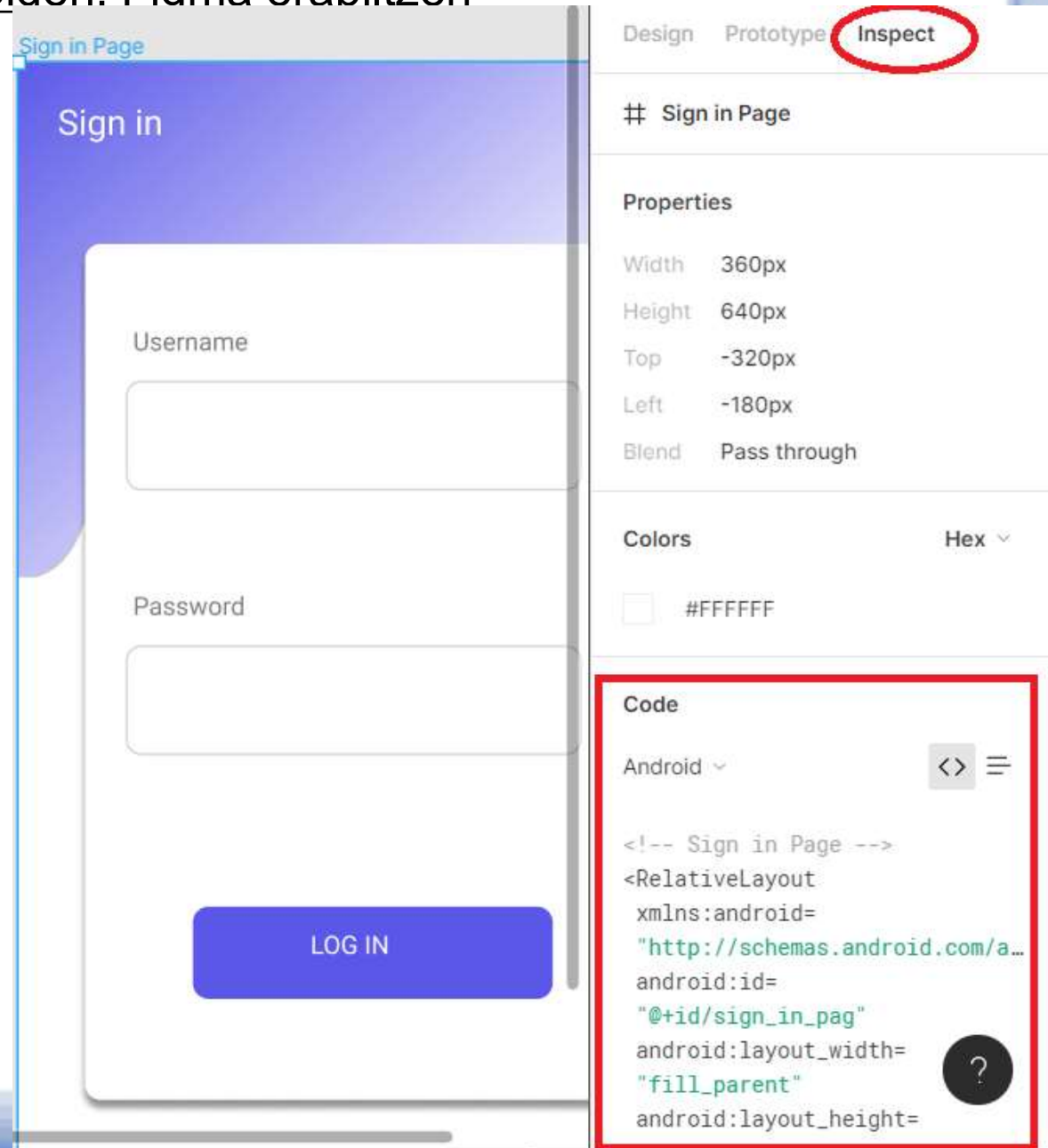
- Exekutatu eta behar diren aldaketak egin:
- Nahi bada, erregistratzeko aukera ere gehitu daiteke



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Figma erabiltzen

Android Studio zabaldu eta  
" FirstAppFigma" deituko dugun  
proiektua egingo dugu, eta  
Figman prestatutako diseinua  
kopiatuko dugu:  
Erabiliko dugu «Figma Inspect»  
Android-era pasatzeko kodea sortzen  
duelako.

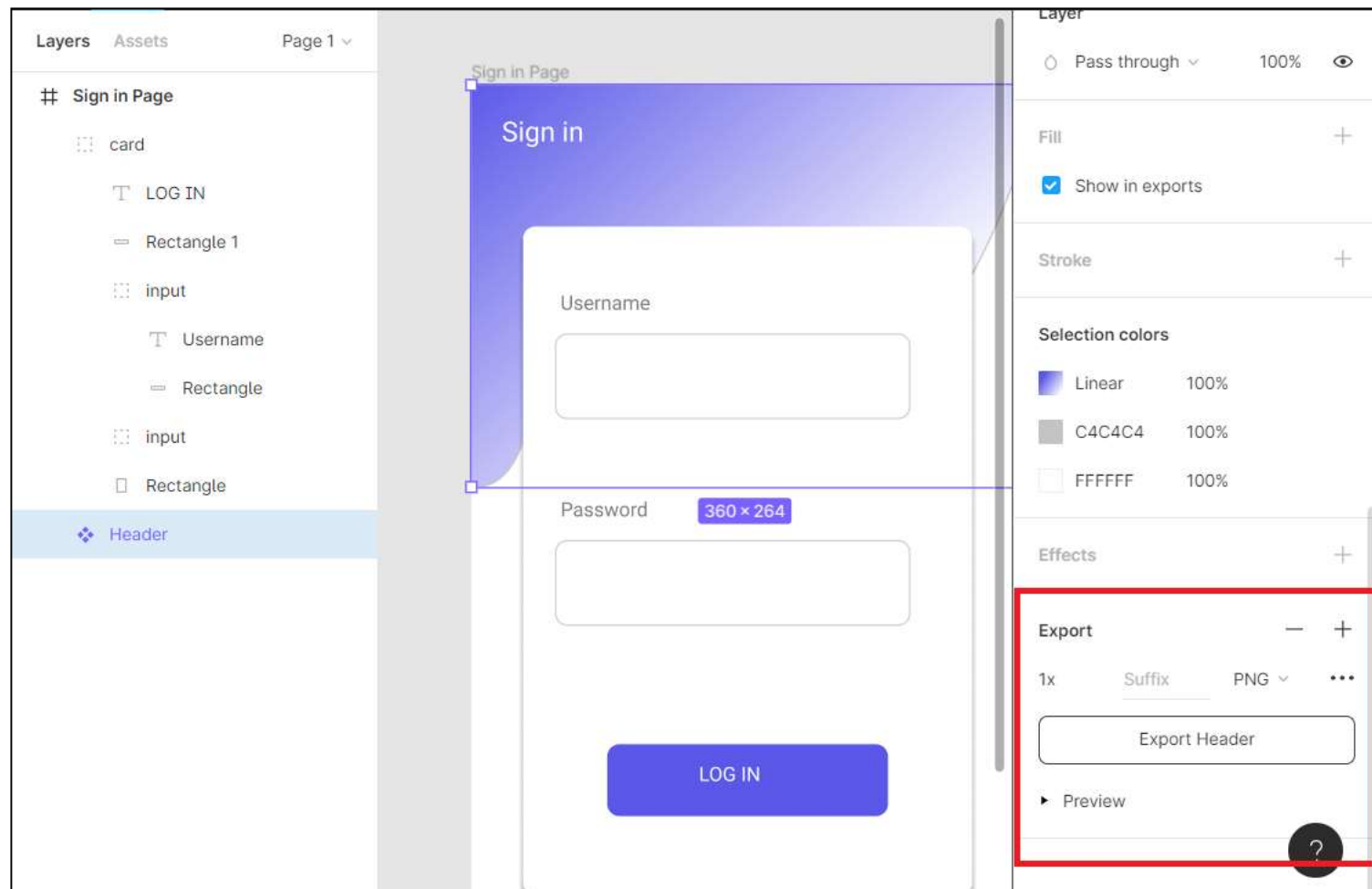


# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Figma erabiltzen

«Header» taldean eskuineko botoiarekin «Crear componente» egin

Eta gero esportatu android aplikazioaren drawable karpetan



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Figma erabiltzen

Inspect pestainatik kodea kopiatu (bakarrik lehenengo zatia) eta android-era eraman (activity\_main.xml)



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <!-- Header -->
10     <RelativeLayout
11         xmlns:android="http://schemas.android.com/apk/res/android"
12         android:id="@+id/header"
13         android:layout_width="360dp"
14         android:layout_height="264dp"
15         android:layout_alignParentLeft="true"
16         android:layout_alignParentTop="true"
17         android:layout_marginTop="-24dp"
18         android:background="@drawable/header"
19     />
20
21 </androidx.constraintlayout.widget.ConstraintLayout>
```

# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Figma erabiltzen

- Berdina beste zatiekin: card, username, rectangle, ... denak ez dira beharrezkoak.
- Fijatu «string» eta «style» batzuk ere behar direla
- Gainera card, «view» barik «RelativeLayout» bezala jarriko dugu beste osagaiak barruan jartzeko.
- Arazoak ematen ditu «constraint» propietateekin, Beraz, «alignParentLeft» eta «alignParentTop» kendu behar dira eta beste batzuk jarri horien ordean.
- **Rectangle --> EditText** aldatu behar dugu, barruan idazteko

```
<!-- card -->
<RelativeLayout
    android:id="@+id/card"
    android:layout_width="295dp"
    android:layout_height="439dp"
    android:background="@drawable/atzealdea_ertza"
    android:layout_marginStart="49dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="89dp" >

    <!-- Username -->
    <TextView
        android:id="@+id/username"
        android:layout_width="129.06dp"
        android:layout_height="27.44dp"
        android:layout_marginStart="48dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="100dp"
        android:text="@string/username"
        android:textAppearance="@style/username"
        android:gravity="top"
    />
```



# Reto 1. Interfaze diseinua XML-ean oinarrituta

## XML Androiden: Figma erabiltzen

Card osagaiari ertza jartzeko «drawable resource» berria sortu daiteke.

```
<shape xmlns:Android="http://schemas.android.com/apk/res/android"
  Android:shape="rectangle" >
  <!-- This is the stroke you want to define -->
  <stroke Android:width="2dp"
    Android:color="#C4C4C4CC"/>

  <!-- Optional, round your corners -->
  <corners Android:bottomLeftRadius="10dp"
    Android:topLeftRadius="10dp"
    Android:bottomRightRadius="10dp"
    Android:topRightRadius="10dp" />

  <!-- Optional, fill the rest of your background with a color or gradient, use transparent if you only want the
border to be displayed-->
  <gradient Android:startColor="#FFFFFFFF"
    Android:endColor="#FFFFFFFF"
    Android:angle="90"/>
</shape>
```

Edo bestela, card osagaia «componente» bezala esportatu daiteke nahi izanez gero. Gure diseinuaren edozein zati esportatu daiteke eta sortzen duen XML-aren background bezala erabili.

# Reto 1. Interfaze diseinua XML-ean oinarrituta

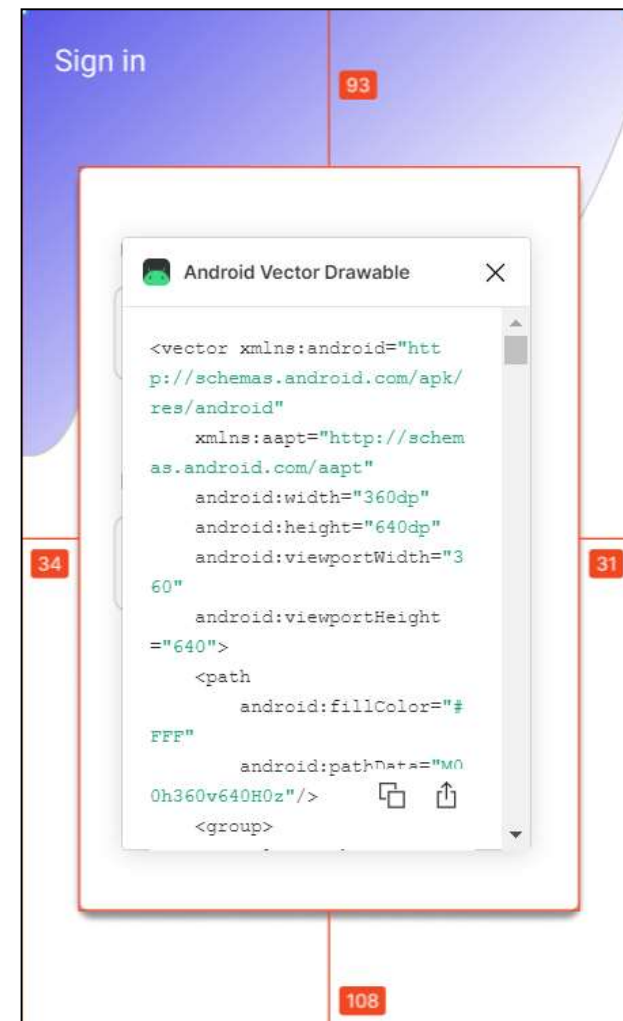
## XML Androiden: Figma erabiltzen

Figman plugin-ak daude. Bat gehitu daiteke: Plugins – Manage plugins – Browse plugins in Community eta «Android Vector Drawable» aukeratu eta instalatuko dugu

Behin instalatuta Plugins menuan agertuko zaigu.

Klik eginda xml guztia fitxategi batean sortzen du.

Irudi bezala erakusteko erabili daiteke.



# XAML

## Estekak:

<https://help.figma.com/hc/en-us/categories/360002042553-Figma-design#Get-started-with-Figma-design>

<https://3ymedia.school/componentes-figma/>

Figman erabiltzeko baliabideak:

<https://figmaelements.com>