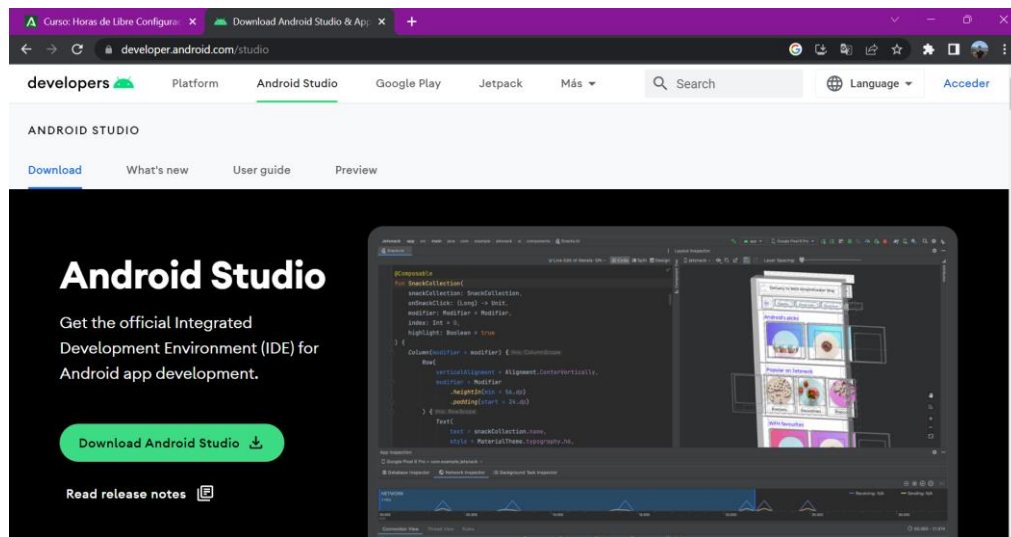


CONTENIDO

Guión 1 – Instalación de Android Studio	2
Guión 2 – Creación del primer login	4
Guión 3 - Pasar mensajes entre activitys.....	6

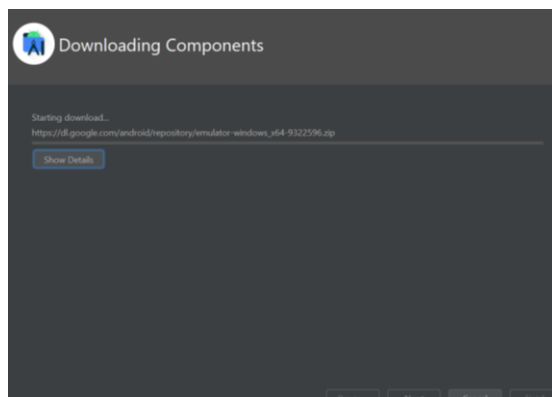
GUIÓN 1 – INSTALACIÓN DE ANDROID STUDIO

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android.

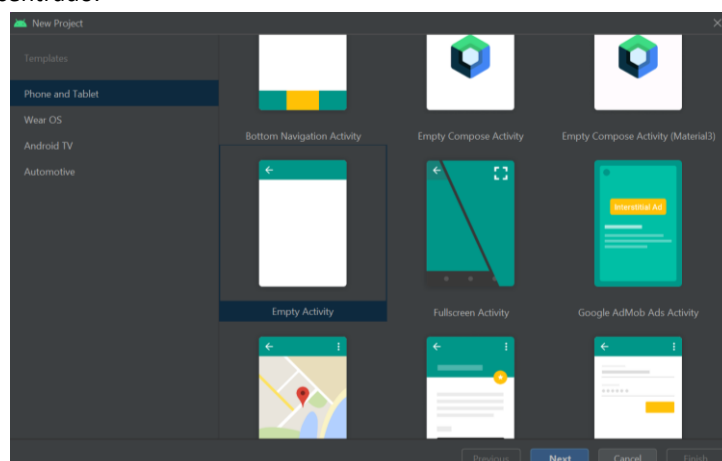


En este primer guion vamos a instalarlo. Vamos a la [página de Android Studio y lo descargamos](https://developer.android.com/studio).

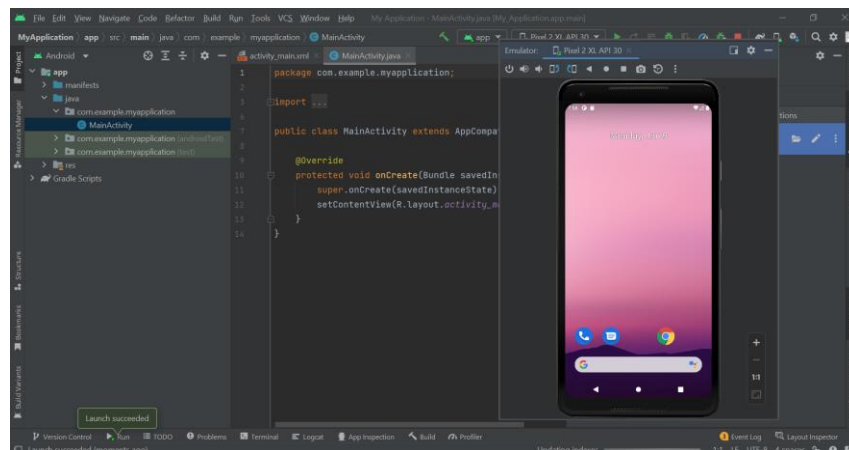
1. Instala Android Studio. Se descargarán los componentes necesarios cuando inicies la instalación.



2. Una vez todo instalado, vamos a crear nuestro Hello World. Crearemos una aplicación simple con un texto centrado.



3. Abrimos el emulador para probar nuestra aplicación.



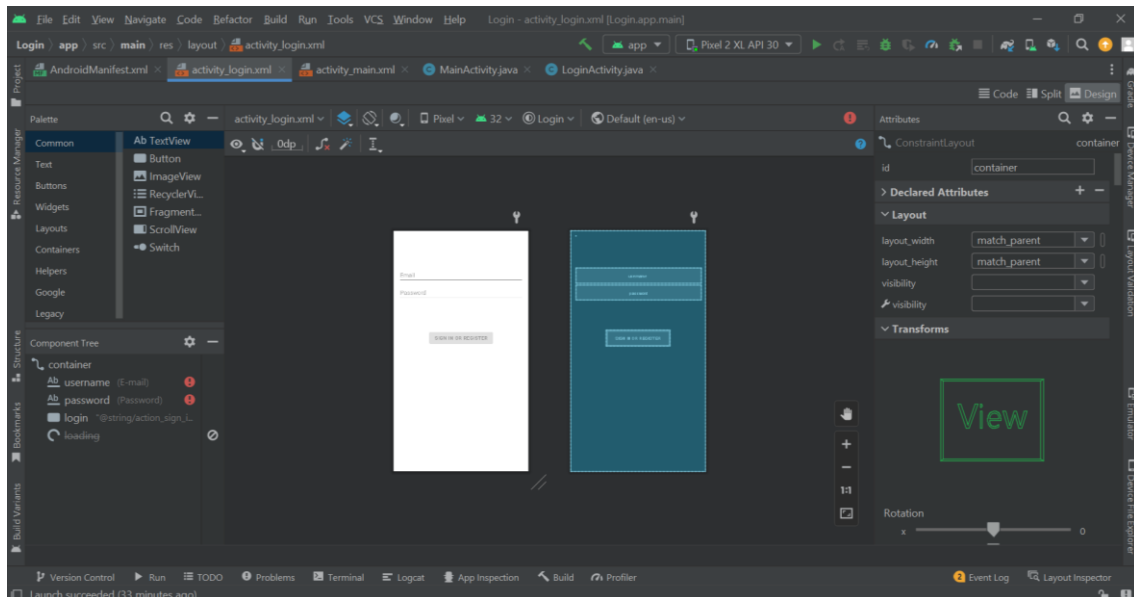
4. La aplicación se mostrará en el emulador.



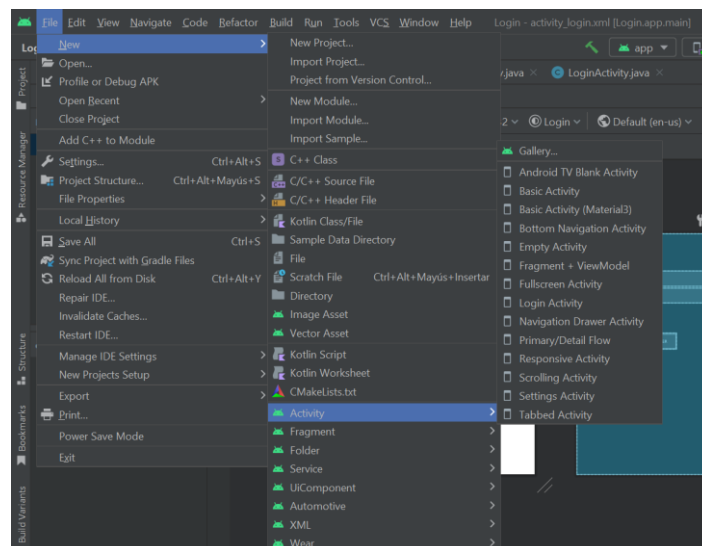
GUIÓN 2 – CREACIÓN DEL PRIMER LOGIN

Hoy vamos a ver como se pueden crear dos “actividades” distintas y enlazar estas. Las actividades son las distintas pantallas que tiene nuestra aplicación y por las que nos vamos moviendo. En nuestro caso vamos a tener la actividad de login donde tendremos el formulario para identificarse, y la actividad principal a la que accederemos a través del botón de login.

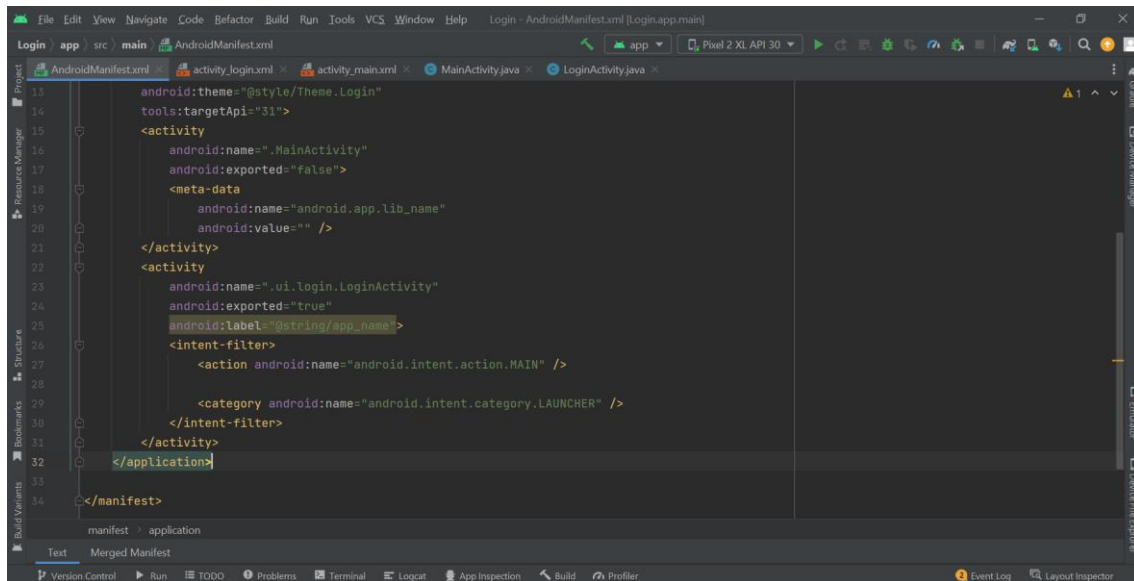
Podemos empezar de dos formas, creando una actividad vacía o creando directamente una actividad de login donde vendrán ya preparados los campos de entrada y el botón.



Una vez tenemos creada la actividad de login, vamos a proceder a crear la principal. Para ello en el menú superior de la izquierda, le damos a file, new, new activity.



En mi caso voy a crear una actividad vacía porque simplemente quiero probar el login, pero veis que se pueden crear diferentes tipos de actividades. Estas se van registrando en el AndroidManifest.xml.



Android Studio la incluye automáticamente si la creamos de la forma explicada. Nótese que la actividad login tiene las líneas de main y launcher, y al iniciar nuestra aplicación se iniciara esta. Una vez tenemos las dos actividades, simplemente tenemos que identificar al botón creado con su ID, y hacer una función que al hacer click en este, ejecute la nueva actividad, la cual sería así:

```

Button btn = (Button)findViewById(R.id.open_activity_button);

btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(MainActivity.this,
MyOtherActivity.class));
    });
});

```

Si hemos creado la primera actividad como login, ya viene creada la vinculación con el id del botón y la función onClick(), la buscamos y añadimos la función startActivity():

```

loginButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        loadingProgressBar.setVisibility(View.VISIBLE);
        loginViewModel.login(usernameEditText.getText().toString(),
            passwordEditText.getText().toString());
        startActivity(new Intent( packageContext: LoginActivity.this, MainActivity.class));
    }
});
}

```

GUIÓN 3 - PASAR MENSAJES ENTRE ACTIVITIES.

Vamos a ver como pasar una cadena de texto de una activity a otra. Para ello vamos a coger el login creado en la sesión anterior, y haremos que en la siguiente pantalla salude al usuario que ha entrado. En mi caso como quiero un nombre de usuario normal, voy a eliminar el campo de email y meter un input de texto normal.

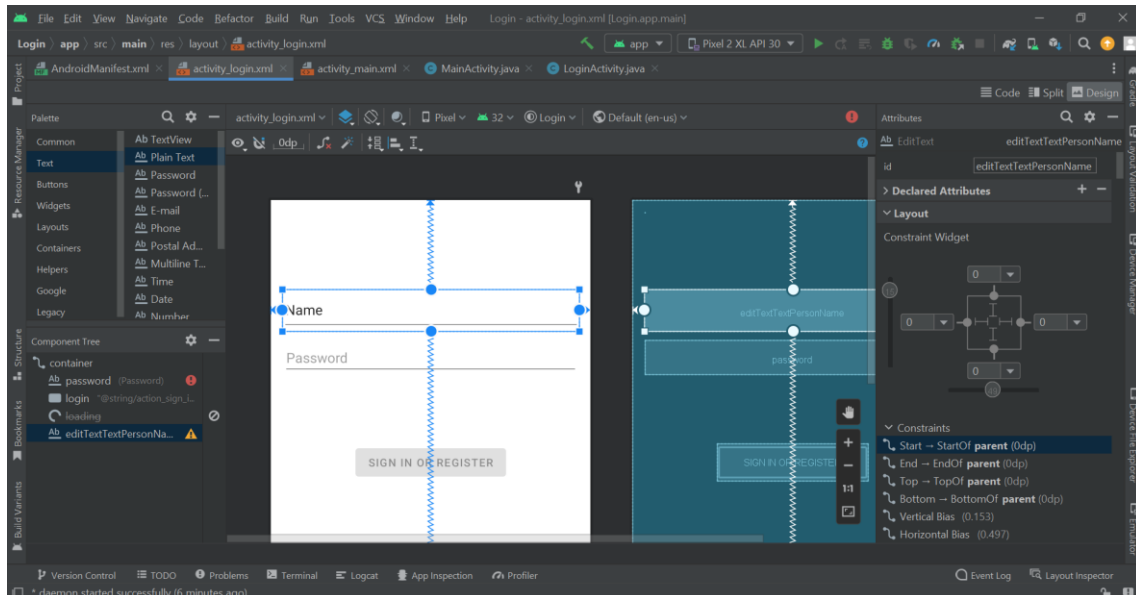


Ilustración 1. En palette metemos del apartado Text un Plain Text.

De id en mi caso voy a llamarlo txtUsuario. Para que salga como placeholder “usuario” vamos a ir al apartado de declared attributes y crear un string usuario.

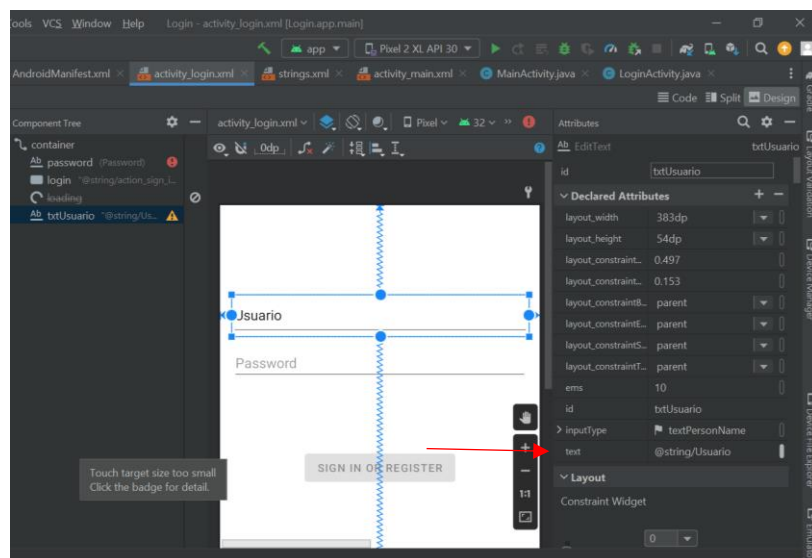


Ilustración 2. En el ovalo que hay al lado del rectangulo para meter texto podemos hacer click para abrir el menu de strings.

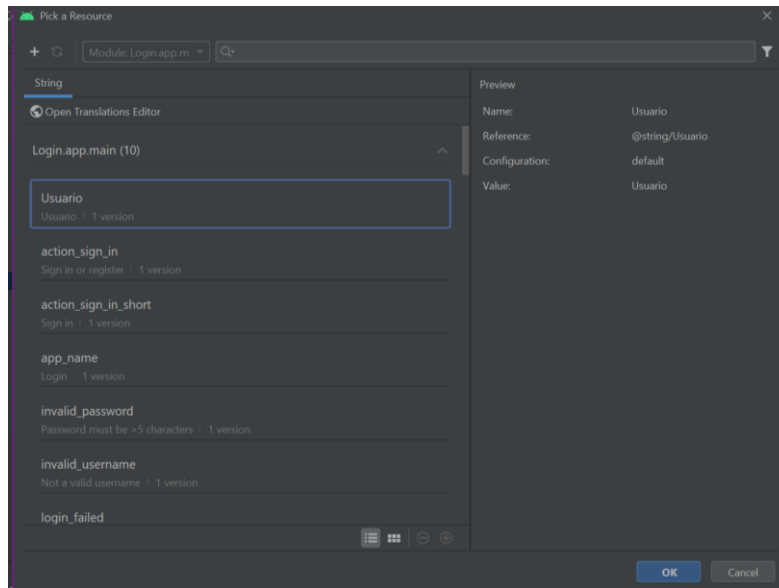


Ilustración 3. Esto es una lista de los strings del programa que almacena en un documento XML.

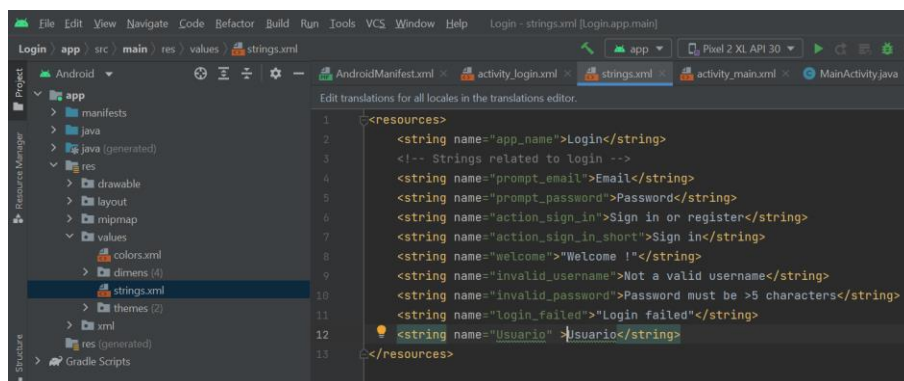
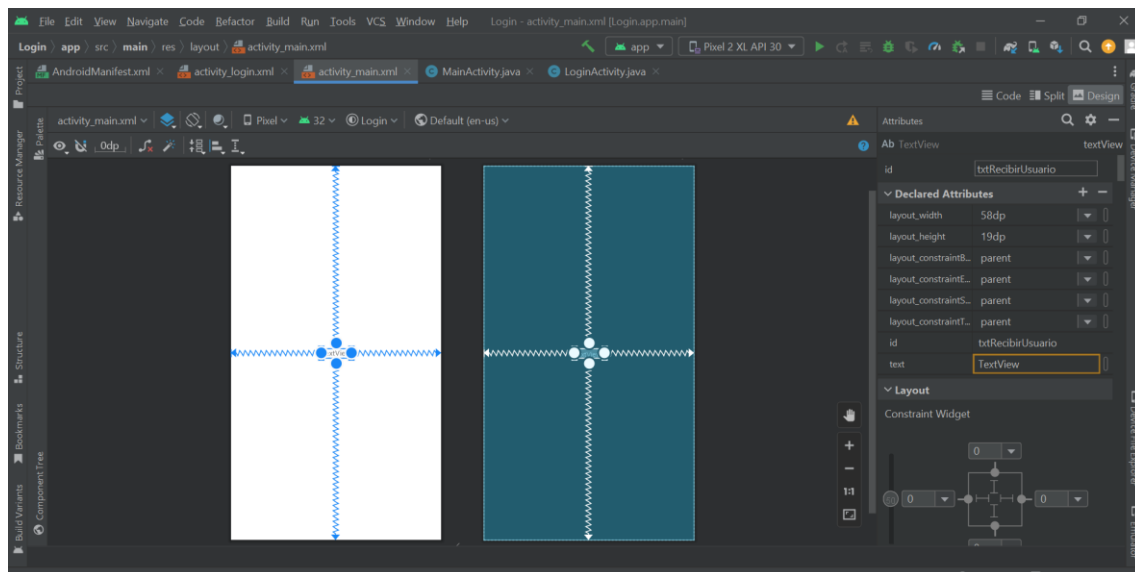


Ilustración 4. Podemos consultar este XML y modificarlo en la carpeta values dentro de res.

Una vez hecho esto ya nos aparecerá usuario. Almacenar los string en un XML puede ser útil por ejemplo si necesitamos hacer una aplicación en varios idiomas, para cambiar todo el texto entre estos. En la segunda pantalla voy a tener un txtRecibirUsuario.



Ahora procedemos a pasar el dato. Recordar que si cambias el input del username, deberéis cambiar también el código java en la clase login para que el objeto java encuentre a ese campo de texto con ese nuevo id.

En la sesión anterior para pasar a la nueva clase creábamos un intent.

```
startActivity(new Intent(LoginActivity.this,
MainActivity.class));
```

Ahora vamos a crear otro intent pero de otra forma.

```
Intent intent = new Intent(LoginActivity.this, MainActivity.class);
intent.putExtra("datos", usernameEditText.getText().toString());
startActivity(intent);
```

Lo que hacemos es asignarle al intent una etiqueta datos que llevara un string asociado a esta. Ahora en la clase a recuperar los datos hacemos lo siguiente:


```
final TextView usuario = findViewById(R.id.txtRecibirUsuario);
String usuarioRecibido = getIntent().getExtras().getString("datos");
usuario.setText(usuarioRecibido);
```

En el código txtRecibirUsuario es mi TextView donde pongo el texto que le paso. Si lo ejecuto vemos que funciona.

