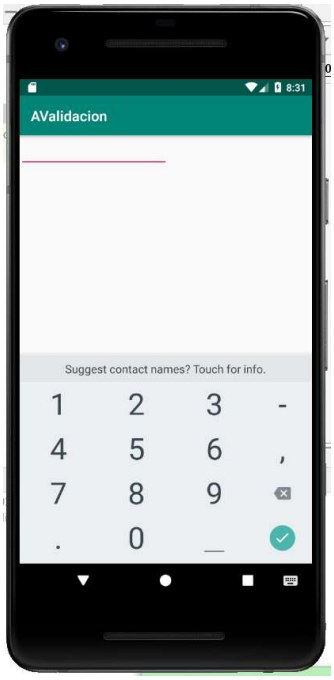
# Utilización de EditText con validación de Campos:

Con la utilización de los EditText (PlainText) vamos a permitir introducir texto en nuestras aplicaciones. A diferencia con los EditText de tipo number ahora cuando ejecutemos nuestra aplicación y vayamos a editar estos campos, en lugar de aparecer el teclado numérico aparecerá teclado de texto.

Con EditText de tipo Number:

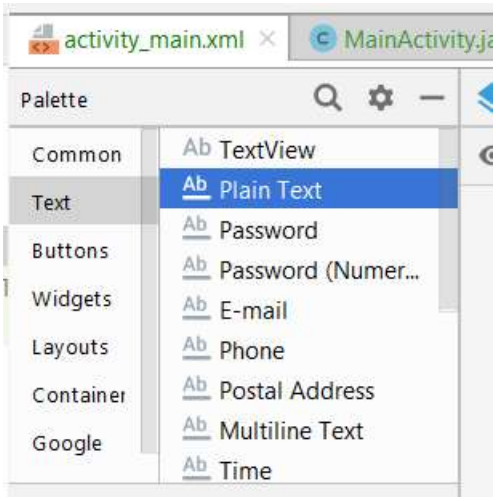


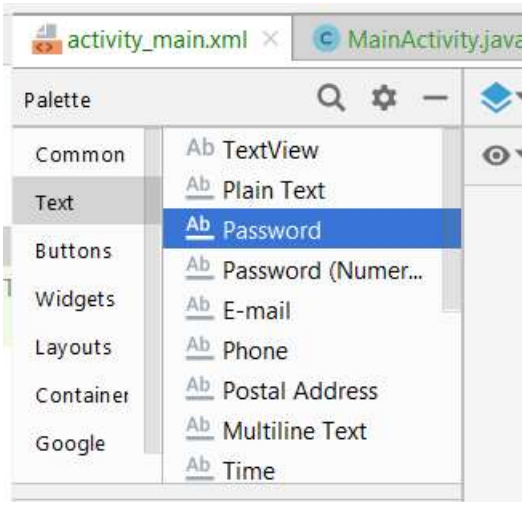

Con EditText de tipo Plain Text:

**Parte grafica:**

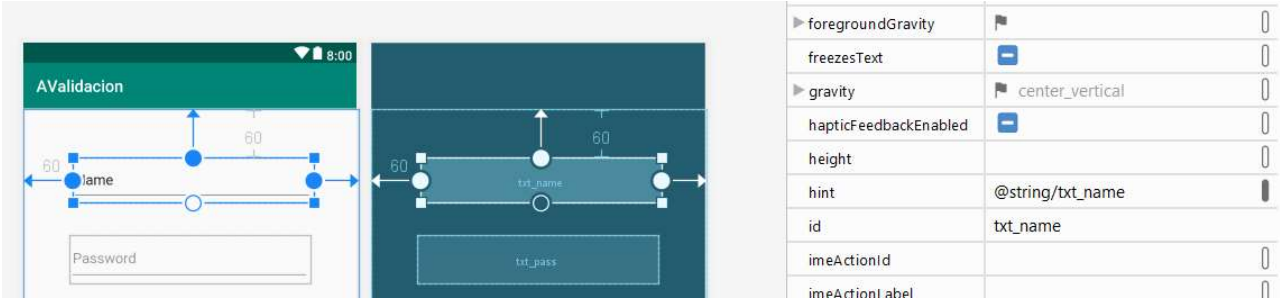
En este ejemplo vamos a diseñar una aplicación que nos solicite usuario y password. Para ello vamos hacer uso de los EditText (Plain Text y Password).

Los EditText de tipo password impide que se visualice lo que estamos escribiendo

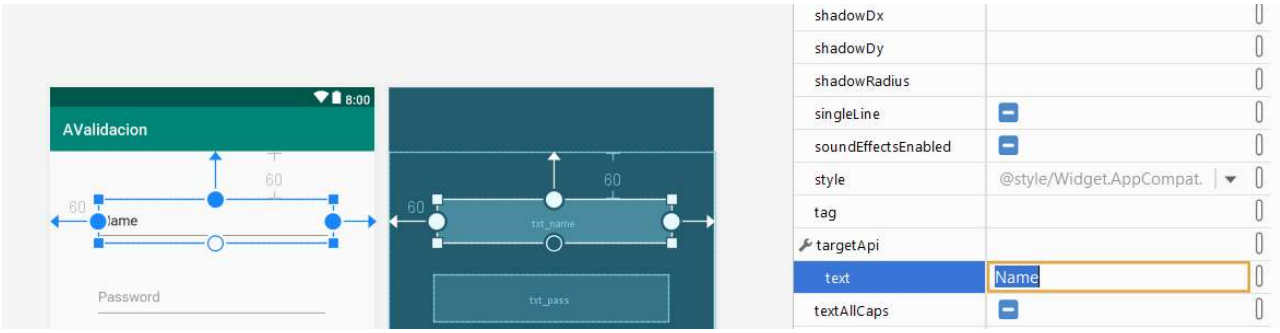


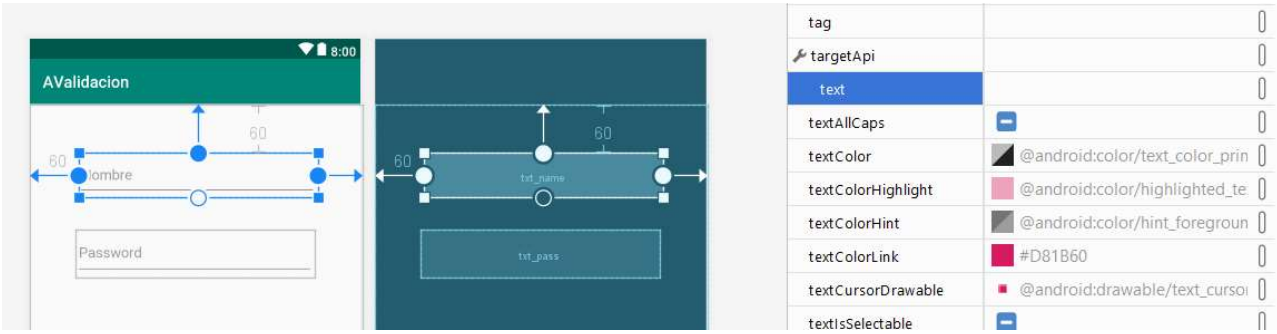

Lo que hay que tener en cuenta con los Plain Text es que cuando asignamos los ***hin*** que se han creado en string.xml aunque los cambiemos en la parte de los atributos, no se modificara en el Plan Text.




Para que esto se modifique, deberemos eliminar la información que aparece en el atributo text



Una vez eliminada ya aparece la información correcta.




Finalizados todos estos cambio la aplicación tendrá una apariencia similar a la imagen.

**Parte lógica:**

Como ya sabemos cuando trabajamos con entornos gráficos en Java, es necesario que indiquemos que componentes vamos a utilizar en nuestro programa.

En este caso vamos a indicar los componentes de tipo TextView.

```
public class MainActivity extends AppCompatActivity

    private EditText et1,et2;
```

Una vez ya tenemos declarados nuestros componentes, ahora toca guardar los valores que vamos a introducir en los EditText.

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

    et1 = (EditText) findViewById(R.id.txt_name);
    et2 = (EditText) findViewById(R.id.txt_pass);
}
```

Ahora deberemos implementar el método al que vamos a llamar con el Button **VALIDAR.**

En este caso simplemente vamos a validar si hemos introducido un usuario o un password.Y si hemos introducido un usuario y un password nos mostrara un mensaje indicando la validación.

Y si en cambio hemos olvidado algunos de los dos campos nos mostrara un mensaje informando que falta información para poder realizar la validación

```
public void Validar (View view)
{
    String nombre = et1.getText().toString();
    String password = et2.getText().toString();

    if (nombre.length()==0)
    {
        Toast.makeText( context: this,  text: "DEBES INTRODUCIR UN NOMBRE", Toast.LENGTH_LONG).show();
    }
    if(password.length()==0)
    {
        Toast.makeText( context: this,  text: "DEBES INTRODUCIR UN PASSWORD", Toast.LENGTH_LONG).show();
    }
    if (nombre.length()!=0 && password.length()!=0)
    {
        Toast.makeText( context: this,  text: "VALIDANDO", Toast.LENGTH_LONG).show();
    }
}
```

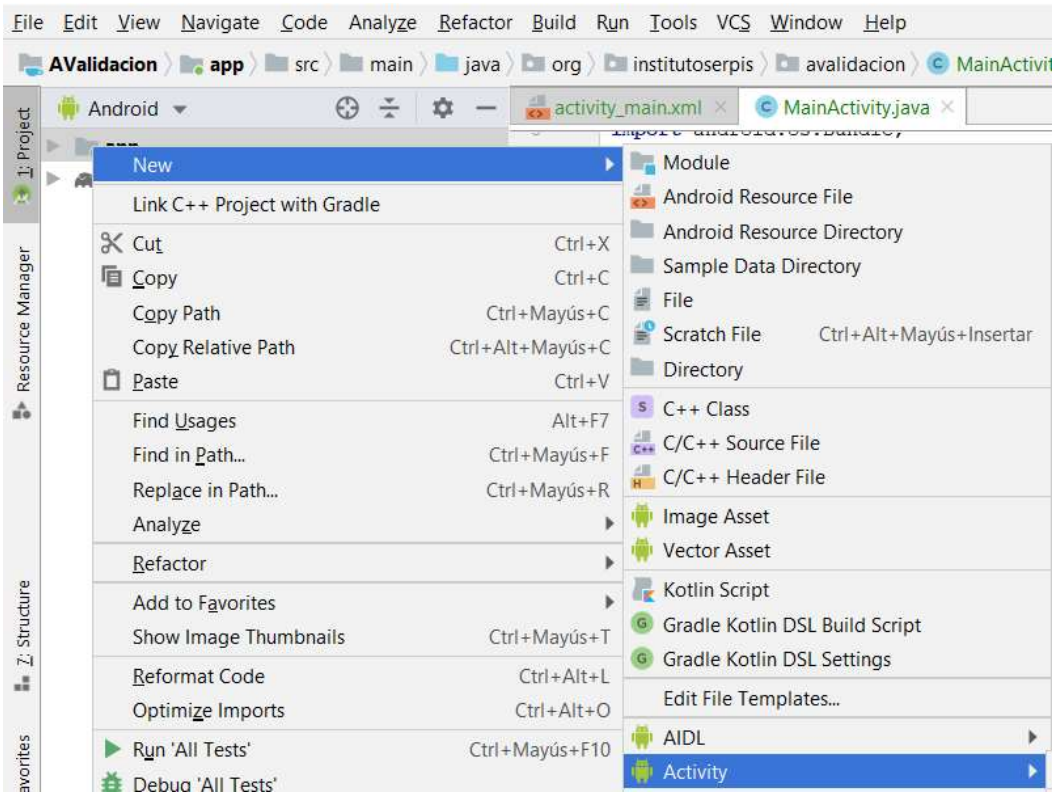
# Como pasar de un Activity a otra:

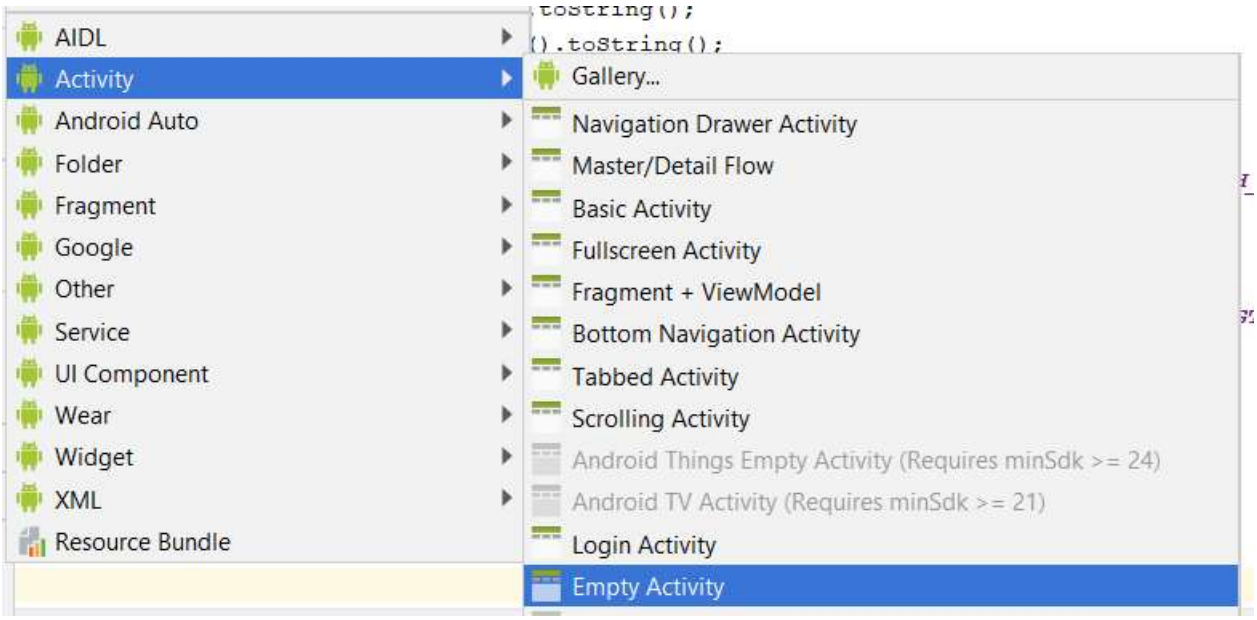
Para navegar a través de las distintas pantallas que puede tener una aplicación, es necesario iniciar una nueva Activity en cada ocasión.

Por cada cambio de pantalla dentro de la aplicación, se debe lanzar o iniciar una nueva Activity, de lo contrario, el cambio de pantalla no se realizará.

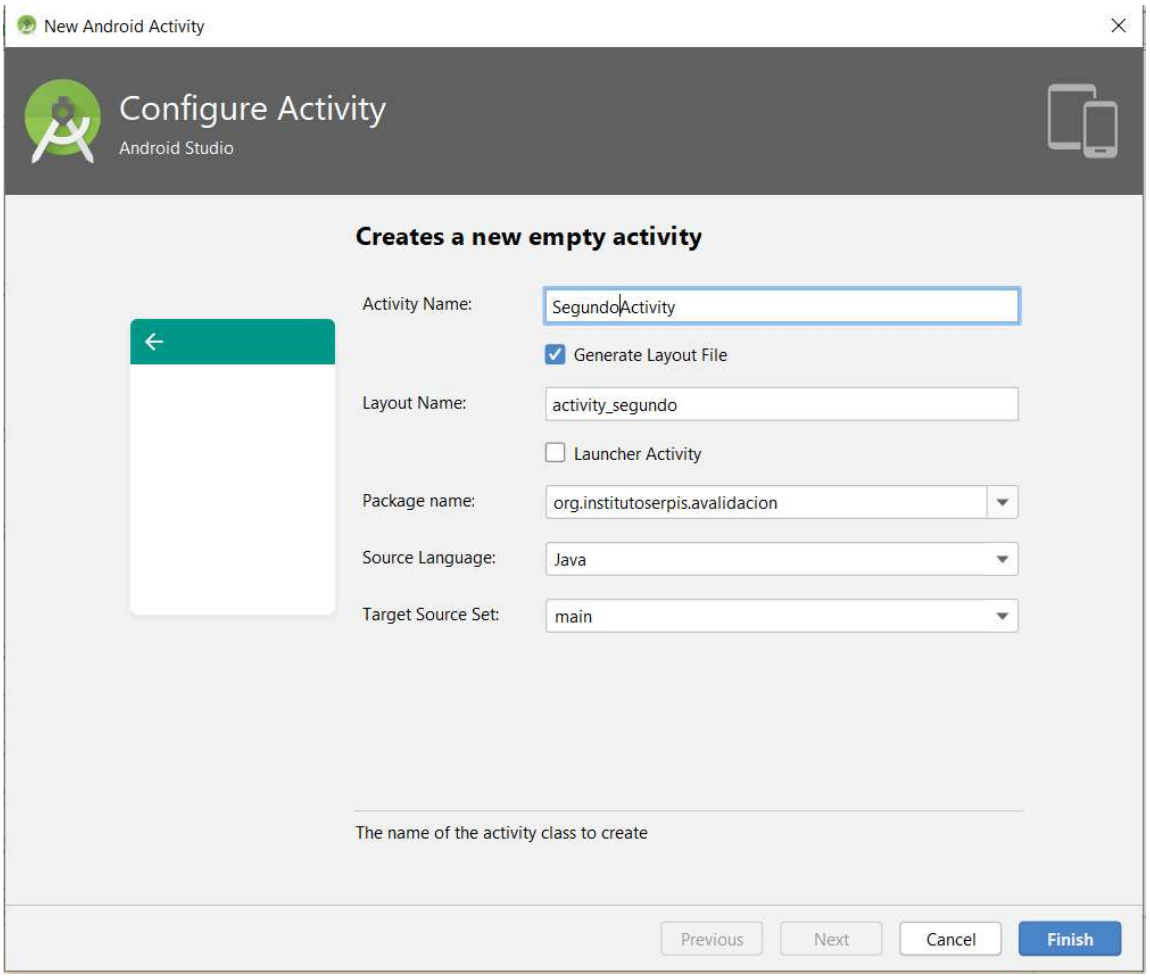
Partiendo de la aplicación anterior vamos a ver como pasar de una activity a otra.

En primer lugar vamos a crear una nueva Empty Activity.Para ello pulsamos con el botón derecho sobre app, luego buscamos Activity y finalmente en Empty Activity.

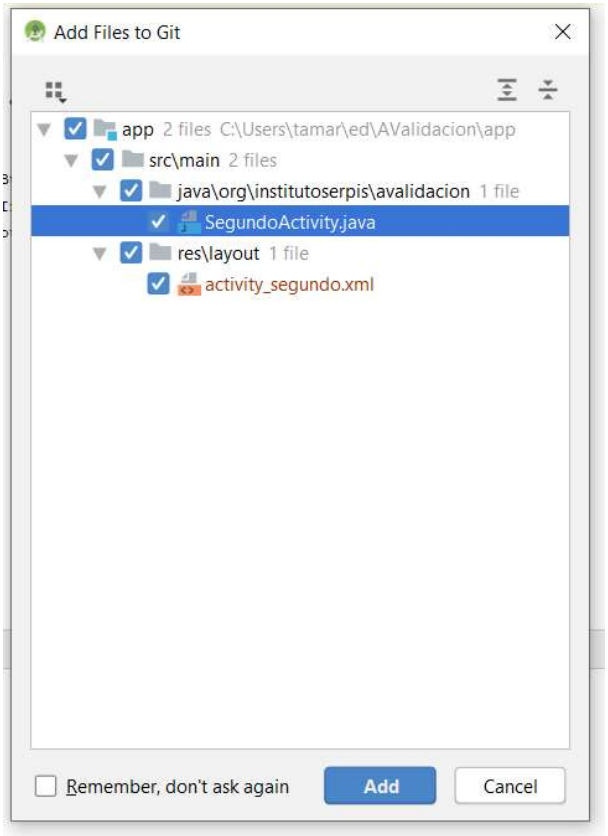





Una vez seleccionada la nueva Empty Activity, vamos a asignarle un nombre. En mi caso le he puesto SegundoActivity (muy original, jijij).



Una vez le damos a Finish puede que nos salga las siguiente ventana, pulsamos en Add y ya se crean nuestros ficheros.




Ahora tendremos dos archivos java y dos archivos xml, pero tendremos unicamente un archivo strings.xml.

**Parte grafica:**

En este caso solo vamos a diseñar la parte grafica de al segunda activity.

En el ejemplo, simplemente he añadido un Button y TextView.


**Parte lógica:**

Para poder realizar el cambio entre Activitys es necesario recurrir a un objeto de tipo ***Intent***.

***Intent*** es un objeto de acción que se puede usar para solicitar una acción de otro componente de la aplicación. Se puede usar ***Intent*** para varias tareas, pero en este ejemplo unicamente lo vamos a usar para iniciar otra activity.

Para crear una ***intent*** basta con crear un objeto de este tipo, la estructura es la siguiente:

```
Intent validar = new Intent(this, SegundoActivity.class );
```

Los parámetros que necesitamos pasarle a este nuevo objeto ***Intent*** son de que activity a que otra activity vamos a pasar.

**`this`** – Activity actual

**`SegundoActivity.class`**-- Activiti a la que queremos llegar

Y finalmente debemos arrancar la activity, para ellos utilizamos la siguiente

```
startActivity(validar);
```

Configuración del Método validar en el MainActivity.java

```
//Método Validar

public void Validar (View view)
{
    String nombre = et1.getText().toString();
    String password = et2.getText().toString();

    if (nombre.length()==0)
    {
        Toast.makeText( context: this,  text: "DEBES INTRODUCIR UN NOMBRE", Toast.LENGTH_LONG).show();
    }
    if(password.length()==0)
    {
        Toast.makeText( context: this,  text: "DEBES INTRODUCIR UN PASSWORD", Toast.LENGTH_LONG).show();
    }
    if (nombre.length()!=0 && password.length()!=0)
    {
        Toast.makeText( context: this,  text: "VALIDANDO", Toast.LENGTH_LONG).show();
        Intent validar = new Intent( packageContext: this, SegundoActivity.class );
        startActivity(validar);

    }
}
```

Configuración del método cerrar en SegundoActivity.java

```
public void Cerrar(View view)
{
    Intent cerrar = new Intent( packageContext: this, MainActivity.class );
    startActivity(cerrar);
}
```

# Como pasar datos o parámetros de un Activity a otra:

Una vez hemos visto como pasar de una activity a otra, ahora vamos a ver como pasar datos o parámetros entre activitys.

Partiendo del ejercicio anterior vamos a implementar como pasar estos datos entre activitys. La parte grafica en este caso no la vamos a modificar, unicamente vamos a realizar modificaciones en la parte lógica.

En este ejemplo lo que vamos a realizar es pasar el dato del nombre del primer activity para mostrarlo en el TextView del segundo activity

Lo primero que debemos realizar es recuperar el valor introducido en el primer TextView para ellos vamos a utilizar el método ***putExtra.***

La configuración del Activity.java del cual queremos coger la información, en este ejemplo el MainActivity.java, es la siguiente:

```
validar.putExtra("Nombre",et1.getText().toString());
```

`validar` - Es la variable que hemos creado de tipo Intent para pasar de una activity a otra.

Al método putExtra le vamos a pasar dos parámetros.

El primero de ellos es con el que vamos a identificar al dato/parámetro que vamos a pasar a la otra activity ,este parámetro debe ser de tipo string e ira entre comillas dobles.

El segundo es la información que queremos enviara a nuestro segundo activity.

La configuración del Activity.java en el cual vamos a mostrar la información, en este ejemplo el SegundoActivity.java, es la que se muestra a continuación:

```
String info=getIntent().getStringExtra("Nombre");
```

Creamos una variable de tipo string para guardar la información que recibimos de la activity anterior. Y utilizamos los métodos ***getIntent()*** y ***getStringExtra ()***, a este segundo método le asignamos el identificador que hemos creado en la primera activity.

Ahora solo nos falta mostrar esa información en el TextView de nuestro SegundoActivity.java

```
tv2.setText("Bienvenida "+info);
```

Configuración MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    private EditText et1,et2;

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

        et1 = (EditText) findViewById(R.id.txt_name);
        et2 = (EditText) findViewById(R.id.txt_pass);
    }
    //Método Validar

    public void Validar (View view)
    {
        String nombre = et1.getText().toString();
        String password = et2.getText().toString();

        if (nombre.length()==0)
        {
            Toast.makeText( context: this,  text: "DEBES INTRODUCIR UN NOMBRE", Toast.LENGTH_LONG).show();
        }
        if(password.length()==0)
        {
            Toast.makeText( context: this,  text: "DEBES INTRODUCIR UN PASSWORD", Toast.LENGTH_LONG).show();
        }
        if (nombre.length()!=0 && password.length()!=0)
        {
            Toast.makeText( context: this,  text: "VALIDANDO", Toast.LENGTH_LONG).show();
            Intent validar = new Intent( packageContext: this, SegundoActivity.class );
            validar.putExtra( name: "Nombre",et1.getText().toString());
            startActivity(validar);

        }
    }
}
```

Configuración SegundoActivity.java

```
public class SegundoActivity extends AppCompatActivity {


    private TextView tv2;

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

        tv2=(TextView)findViewById(R.id.txt_Bienvenido);
        String info=getIntent().getStringExtra( name: "Nombre");
        tv2.setText("Bienvenida "+info);

    }

//Método Cerrar

    public void Cerrar(View view)
    {
        Intent cerrar = new Intent( packageContext: this, MainActivity.class );
        startActivity(cerrar);
    }

}
```

Cuando ejecutamos nuestra aplicación el Resultado es el siguiente: