

## PRACTICA 08: Paso de parámetros y datos entre activities a través de Bundle

CENTRO DE CIENCIAS BASICAS

I.S.C.

9º. SEMESTRE, GRUPO C

OPTATIVA, (PROGRAMACION DE MOVILES)

PROFESOR: M.C. MIGUEL ANGEL ORTIZ ESPARZA.

Octubre 2017.

Como vimos en la práctica anterior es importante las transiciones ordenadas en el sistema, que van de pantalla en pantalla, otra cosa importante dentro de un sistema no solo es que se puedan llamar las actividades, sino también que estas puedan pasar datos o parámetros de una a otra con el objeto de mejorar la interacción con el usuario, evitando la monotonía de los sistemas centrados.

Por ello en la siguiente practica veremos el paso de datos o parámetros entre activities. Basándonos en la práctica anterior vamos a agregar algunas líneas de texto y algunas vistas en los layouts existentes para completar esta practica

### Objetivo.

Conocer la importancia de los extras y del paso de datos o parámetros entre las activities.

### Procedimiento.

1. Modificamos el XML del activity\_main.xml y agregamos el siguiente elemento de la siguiente forma.

```
<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/enviar"/>
```

2. También modificamos el archivo strings.xml el cual tiene ahora la siguiente forma.

```
<resources>
    <string name="app_name">Actividad Principal</string>
    <string name="boton">Lanzar la segunda actividad</string>
    <string name="title_activity_second">Segunda Actividad</string>
    <string name="enviar">Texto para enviar</string>
</resources>
```

3. Ahora vamos a el archivo MainActivity.java y agregamos las siguientes líneas de código las cuales comprenden la identificación del TextView que hemos creado anteriormente.

- o `EditText editText;`
- o `editText= (EditText) findViewById(R.id.editText);`

El código estaría de la siguiente forma después de la modificación.

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{
    Button boton;
    EditText editText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        boton = (Button) findViewById(R.id.boton);
        editText= (EditText) findViewById(R.id.editText);
        boton.setOnClickListener(this);
    }
```

4. Ahora vamos a modificar dentro de onClick (View v) las siguientes líneas de código que a continuación se explican. Primero vamos a obtener el texto en forma de cadena desde el TextView anteriormente creado con la siguiente línea de código, la cual ya la convertimos de una vez en tipo cadena de texto.
- o `String dato = editText.getText().toString();`

# Universidad Autónoma de Aguascalientes.

5. Ahora con el putExtra vamos a enviar el dato que se obtuvo desde el EditText, primero debemos colocar el nombre del dato a enviar en este caso "MENSAJE", con el dato a enviar, el cual es el string anteriormente creado llamado "dato", lo cual nos queda un código como el presentado a continuación.

```
@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.boton:
            String dato = editText.getText().toString();
            Intent siguiente = new Intent(MainActivity.this, SecondActivity.class);
            siguiente.putExtra("MENSAJE",dato);
            startActivity(siguiente);
            break;
    }
}
```

6. Ahora en el lado de la segunda Activity vamos a crear un Bundle denominado extras, el cual funciona a través de capturar el intent enviado desde la actividad principal

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

    Intent intent = getIntent();
    Bundle extras = intent.getExtras();
}
```

7. También vamos a crear ponerle un id=texto al TextView que habíamos creado anteriormente en la second activity con las siguientes líneas de código.

```
<TextView
    android:id="@+id/texto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:layout_editor_absoluteY="0dp"
    tools:layout_editor_absoluteX="8dp"
    android:layout_weight="0.06"
    android:text="Segunda Actividad"
    android:textSize="40dp"
    android:gravity="center"/>
```

8. Lo ligamos al SecondActivity.java de la siguiente forma quedando el código de la siguiente forma.

```
public class SecondActivity extends AppCompatActivity {
    TextView texto;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        texto= (TextView) findViewById(R.id.texto);

        Intent intent = getIntent();
        Bundle extras = intent.getExtras();
    }
}
```

9. Ahora vamos a validar que el dato que viene de la MainActivity no venga vacío con las siguientes líneas de código.

```
        if(extras!=null){  
            String dato =extras.getString("MENSAJE");  
            texto.setText(dato);  
        }  
    }
```

10. Con esto validamos que el texto no tenga vacío, además de corroborar que el tipo de dato que estamos accediendo es de tipo string.