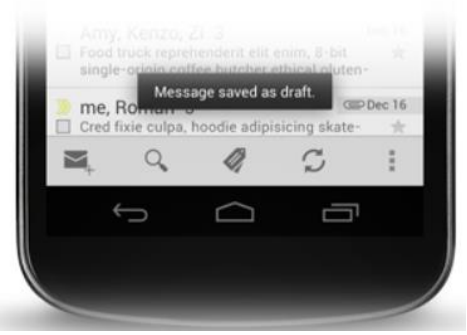
INDICE

1.1. Introducción

- Un **Toast** proporciona un simple feedback de algo que sucede en la aplicación a través de un pequeño mensaje emergente (popup).
- Se llama "Tostada" porque aparece en la pantalla de la misma forma en que las tostadas saltan en la tostadora cuando están hechas.



- Para usar el Toast:

```
1 Context context = getApplicationContext();
2 CharSequence text = "Hello toast!";
3 int duration = Toast.LENGTH_SHORT;
4
5 Toast toast = Toast.makeText(context, text, duration);
6 toast.show();
```

- **Context**: es un interface que permite acceder a los recursos de nuestra aplicación

  Para acceder a la interface podemos emplear.

- La palabra this se hace referencia a la activity (por ejemplo, dentro de una interface anónima OnClickListener asociadada a un botón, this hace referencia a un botón, no a la activity y no podríamos emplear esa palabra.

```
public class MainActivity extends AppCompatActivity {


    @Override
    protected void onStart() {
        super.onStart();

        Toast t = Toast.makeText(this,"START",Toast.LENGTH_LONG);
    }

}
```

- **Nombre de la activity.this.** Por ejemplo, dentro de un botón gestionando el evento click:

```
Button btn = findViewById(R.id.btnLanzar_Act_Main);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(MainActivity.this,"PULSADO BO-
TON",Toast.LENGTH_LONG).show();

            }
        });
```

- Empleando el método **getApplicationContext(),** por ejemplo:

```
Button btn = findViewById(R.id.btnLanzar_Act_Main);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(getApplicationContext(),"PULSADO BO-
TON",Toast.LENGTH_LONG).show();

            }
        });
```

- **Duración:**
  - ✓ **LENGTH_SHORT**: constante que equivale a 2 segundos.
  - ✓ **LENGTH_LONG**: constante que equivale a 3,5 segundos.

- Las líneas 5 y 6 anteriores pueden ser substituidas por la siguiente:

```
Toast.makeText(context, text, duration).show();
```

- **Referencias:**
  - ✓ Clase **Toast**: https://developer.android.com/reference/android/widget/Toast

- ✓ Introducción: https://developer.android.com/guide/topics/ui/notifiers/toasts

1.2. Caso práctico

- Partimos que ya tenemos creado el proyecto inicial.

  Si no lo tenemos creado antes, crear un paquete de nome **UI** como un subpaquete de tu paquete principal.

  Dentro do paquete UI, crearemos un nuevo paquete de nombre: **Toasts**.

- Dentro del paquete **Toasts** crear una nueva 'Empty Activity' de nombre: **UD02_01_ Toast** de tipo Launcher.

  Modificar el archivo **AndroidManifiest.xml** y añade un label a la activity como ya vimos en la creación del proyecto base.

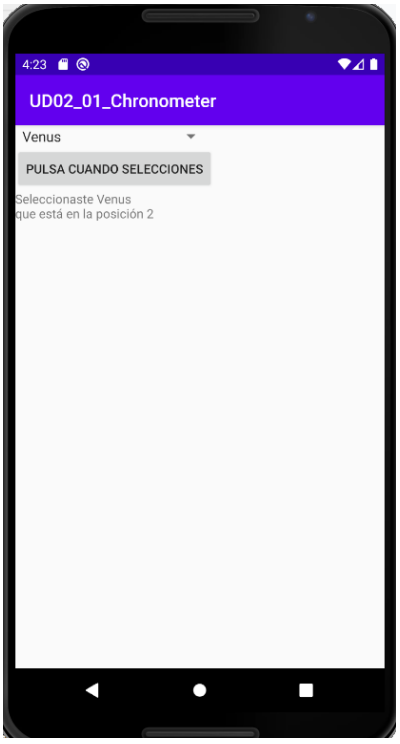
- Tomando como base la primera versión de la aplicación de los planetas: U2_12_Spinner .
- Justo ahora, la nueva aplicación tiene una entrada más en la parte superior de **Select a Planet** Spinner.

Toast Planetas

Contenido de Spinner.

No se seleccionó ningún planeta (esta seleccionado el ítem 0 del Spinner). Pulsar botón.

Aparece una advertencia que indica que no se ha seleccionado ningún planeta (está selecciona la entrada 0 del Spinner)



Si seleccionamos un planeta funciona como se esperaba, solo que ahora la posición cambio con respecto a la aplicación original.

Archivos de recursos: array_planetas.xml

- Copiamos el fichero **array_planetas.xml** ubicado en **/res/values/** de una aplicación a otra.
- Añadimos un ítem al principio:

```
<?xml version="1.0" encoding="utf-8"?>

<resources>
    <string-array name="planetas_UD02_01_spinner">
        <item>Mercurio</item>
        <item>Venus</item>
        <item>Tierra</item>
        <item>Marte</item>
        <item>Júpiter</item>
        <item>Saturno</item>
        <item>Urano</item>
        <item>Neptuno</item>
    </string-array>

    <string-array name="planetas_UD02_01_toast">
        <item>Selecciona un planeta</item>
        <item>Mercurio</item>
        <item>Venus</item>
        <item>Tierra</item>
        <item>Marte</item>
        <item>Júpiter</item>
        <item>Saturno</item>
        <item>Urano</item>
```

```
        <item>Neptuno</item>
    </string-array>
</resources>
```

1.2.1. XML del Layout

- Es igual al de la aplicación original, cambiando la fuente de datos del spinner y actualizando los id's de los controles. Observar cómo se crea el control Chronometer y los métodos a los que llaman los botones.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".UD02_01_Toast">

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >

        <Spinner
            android:id="@+id/spnPlanetas_UD02_01_Toast"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:entries="@array/planetas_UD02_01_toast" />

        <Button
            android:id="@+id/btnSeleccion_UD02_01_Toast"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="onButtonClick"
            android:text="Pulsa cuando selecciones" />

        <TextView
            android:id="@+id/tvResultado_UD02_01_Toast"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Selecciona un planeta" />

    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

1.2.2. Código Java

```
package com.example.ud02_01_chronometer;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;



public class UD02_01_Toast extends AppCompatActivity {

    public void onButtonClick(View v) {
        Spinner spinPlanetas = (Spinner) findViewById(R.id.spnPlanetas_UD02_01_Toast);
        TextView lblResultado = (TextView) findViewById(R.id.tvResultado_UD02_01_Toast);

        if (spinPlanetas.getSelectedItemId() == 0) {
            Toast.makeText(this, "Por favor selecciona un planeta",Toast.LENGTH_LONG).show();
            lblResultado.setText("");
        } else

            lblResultado.setText("Seleccionaste "
                    + spinPlanetas.getSelectedItem()
                    + "\nque está en la posición "
                    + spinPlanetas.getSelectedItemId());
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u_d02_01__toast);
    }
}
```

- **Línea 18:** Comprobamos si la posición seleccionada en el Spinner es a 0 de ser así…
- **Línea 19:** realizamos un Toast.
- **Línea 20:** Ponemos la etiqueta en vacío.

1.3. Personalizando el Toast

- Toast es un objeto de la clase Toast que se crea cuando llamamos a makeToast.

  Una vez creado podemos acceder a sus propiedades y métodos.

- Por ejemplo:
  - ✓ setGravity (int gravity, int xOffset, int yOffset): Cambiamos la posición predeterminada en la que aparece el Toast. Debemos hacer uso de la clase Gravity.
  - ✓ setView(View view): Podemos establecer que se visualiza un view definido por nos.

- ✓ setText (int resId): Podemos hacer uso de un recurso de tipo 'string' haciendo referencia a él de la forma: R.string.resource_name

➢ Veamos un ejemplo, basado en el mismo código que el ejemplo anterior, en el que crearemos una vista TextView que asociaremos con el Toast y también modificaremos la posición en la que aparece el mensaje:

➢ El código modificado:

```
package com.example.ud02_01_chronometer;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;



public class UD02_01_Toast extends AppCompatActivity {

    public void onButtonClick(View v) {
        Spinner spinPlanetas = (Spinner) findViewById(R.id.spnPlanetas_UD02_01_Toast);
        TextView lblResultado = (TextView) findViewById(R.id.tvResultado_UD02_01_Toast);

        if (spinPlanetas.getSelectedItemId() == 0) {
            Toast toast = Toast.makeText(this, "",Toast.LENGTH_LONG); //Creamos un objeto Toast
            TextView viewToast = new TextView(this);    // Creamos un view que vamos a asociar-
lo al Toast
                                                        // en vez de una cadena de tex-
to.
                                                        // Podría ser cualquier view
            viewToast.setText("Por favor selecciona un planeta");
```

```
            viewToast.setTextColor(Color.RED);

            toast.setGravity(Gravity.TOP|Gravity.RIGHT,10,10);  // Cambiamos las propiedades
del toast
            toast.setView(viewToast);

            toast.show();   // Mostramos el toast
            lblResultado.setText("");
        } else

            lblResultado.setText("Seleccionaste "
                    + spinPlanetas.getSelectedItem()
                    + "\nque está en la posición "
                    + spinPlanetas.getSelectedItemId());
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u_d02_01__toast);
    }
}
```

1.4. Empleando cadenas definidas en /res/values

- Como ya vimos en los TextViews, las cadenas que se muestran en un Toast no deben ser escritas directamente en el código Java, si no que deben ser referenciadas como un recurso de tipo cadena definido en algún archivo /res/values