



Elementos de la interfaz, navegación e interacción

Material complementario

/* Componentes visuales */

EditText para material design

Mejoramiento de la interfaz

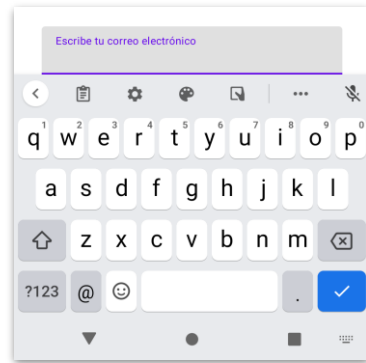
TextInputLayout: Estiliza los campos de textos con etiquetas flotantes.

TextInputEditText: Es una subclase de EditText diseñada para ser usada dentro de un TextInputLayout. El atributo `inputType` permite definir el tipo de teclado que se utilizará.

```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/text_input_layout_email"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginEnd="32dp"
    android:hint="@string/name"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress" />

</com.google.android.material.textfield.TextInputLayout>
```

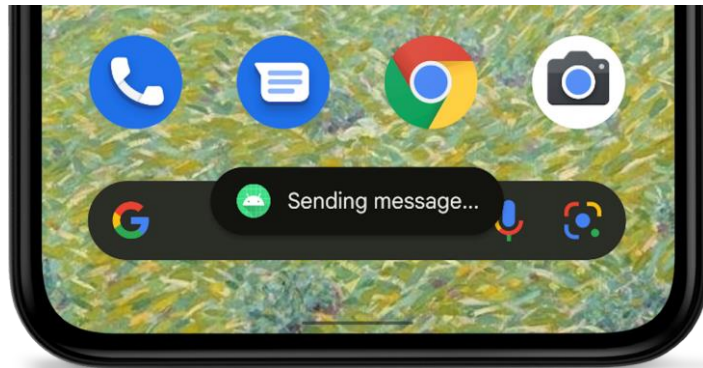


/* Mensajes emergentes */

Toast

Es un aviso que entrega información sobre una acción en una pequeña ventana emergente que desaparece después de un tiempo de espera.

Por ejemplo, si se hace clic en la opción **Enviar** de un correo electrónico, se activará el aviso "Enviando mensaje...", como se muestra en la siguiente captura de pantalla:



Toast

Para crear un mensaje Toast utilizamos el método `makeText` que recibe:

- el contexto
- el texto a mostrar:
 - como texto
 - como un ID de un recurso string.
- La duración del mensaje puede ser:
 - `Toast.LENGTH_SHORT`
 - `Toast.LENGTH_LONG`

```
Context context = getContext();
CharSequence text = "Este es un mensaje para el usuario";
int duration = Toast.LENGTH_SHORT;

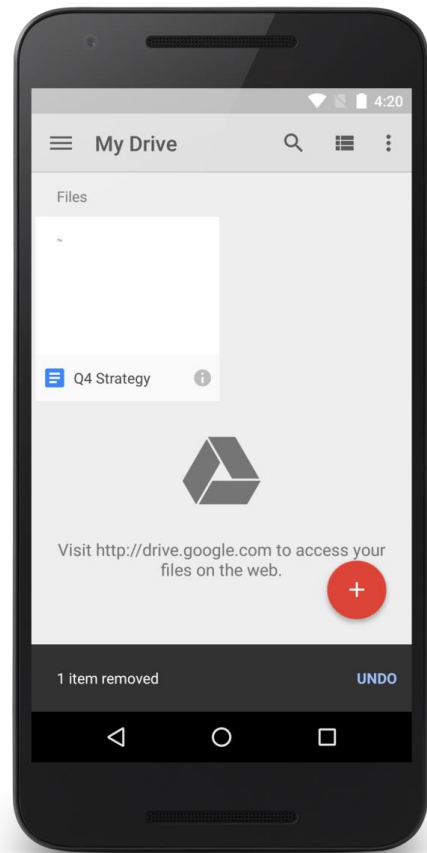
Toast toast = Toast.makeText(context, text, duration).show();
```

Importante: Luego de creado el toast se debe llamar al método `show()` para que el mensaje se muestre al usuario.

Snackbar

★ **Nota:** La clase [Snackbar](#) sustituye a [Toast](#). Si bien [Toast](#) sigue siendo compatible por el momento, [Snackbar](#) es la forma preferida actual de mostrar mensajes breves y transitorios al usuario.

Snackbar permite mostrar un mensaje rápido al usuario y opcionalmente se pueden agregar acciones rápidas



Snackbar

Para crear un mensaje Snackbar utilizamos el método `makeText` que recibe:

- La vista a la que se adjunta el Snackbar.
- el texto a mostrar:
 - como texto
 - como un ID de un recurso string.
- La duración del mensaje puede ser:
 - `Snackbar.LENGTH_SHORT`
 - `Toast.LENGTH_LONG`

```
View view = getView();
String text = "Este es un mensaje para el usuario";
int duration = Snackbar.LENGTH_LONG;

Snackbar.make(view, text, duration).show();
```

Importante: Luego de creado el snackbar se debe llamar al método `show()` para que el mensaje se muestre al usuario.

{desafío}
latam_

*Academia de
talentos digitales*

