Cómo especificar el tipo de método de entrada

Cada campo de texto espera un cierto tipo de entrada de texto, como una dirección de correo electrónico, un número de teléfono o simplemente texto sin formato. Por lo tanto, es importante que especifiques el tipo de entrada para cada campo de texto de tu app, de manera que el sistema muestre el método de entrada de software apropiado (como un teclado en pantalla).

Además del tipo de botones disponibles con un método de entrada, debes especificar comportamientos como si el método de entrada proporciona sugerencias ortográficas, escribe mayúsculas en oraciones nuevas y reemplaza el botón de retorno de carro por un botón de acción como **Listo** o **Siguiente**. En esta lección, se muestra cómo especificar estas características.

Cómo especificar el tipo de teclado

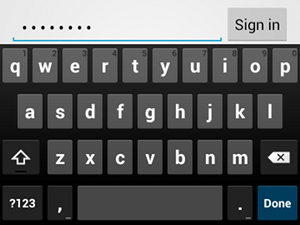
Siempre debes declarar el método de entrada para tus campos de texto al agregar el atributo [android:inputType](https://developer.android.com/reference/android/widget/TextView.html#attr_android:inputType) al elemento [<EditText>](https://developer.android.com/reference/android/widget/EditText.html).



**Figura 1:** Tipo de entrada phone

Por ejemplo, si quieres usar un método de entrada para ingresar un número de teléfono, usa el valor "phone":

    <EditText  
        android:id="@+id/phone"  
        android:layout\_width="fill\_parent"  
        android:layout\_height="wrap\_content"  
        android:hint="@string/phone\_hint"  
        android:inputType="phone" />



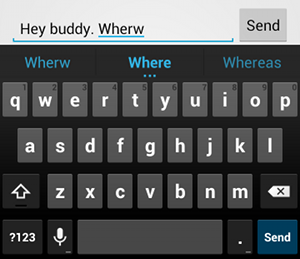
**Figura 2:** Tipo de entrada textPassword

En cambio, si el campo de texto es de una contraseña, usa el valor "textPassword" para que el campo de texto oculte la entrada del usuario:

    <EditText  
        android:id="@+id/password"  
        android:hint="@string/password\_hint"  
        android:inputType="textPassword"  
        ... />

Hay varios valores posibles documentados con el atributo [android:inputType](https://developer.android.com/reference/android/widget/TextView.html#attr_android:inputType) y se pueden combinar algunos de los valores para especificar la apariencia del método de entrada y comportamientos adicionales.

Cómo habilitar sugerencias de ortografía y otros comportamientos



**Figura 3:** Al agregar textAutoCorrect, se ofrece corrección automática de errores de ortografía.

El atributo [android:inputType](https://developer.android.com/reference/android/widget/TextView.html#attr_android:inputType) te permite especificar varios comportamientos para el método de entrada. Lo más importante es que si tu campo de texto está destinado a la entrada de texto básica (como un mensaje de texto), debes habilitar la corrección de ortografía automática con el valor "textAutoCorrect".

Puedes combinar diferentes comportamientos y estilos de métodos de entrada con el atributo [android:inputType](https://developer.android.com/reference/android/widget/TextView.html#attr_android:inputType). Por ejemplo, a continuación se explica cómo crear un campo de texto que capitaliza la primera palabra de una oración y también corrige automáticamente los errores ortográficos:

    <EditText  
        android:id="@+id/message"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:inputType=  
            "textCapSentences|textAutoCorrect"  
        ... />

Cómo especificar la acción del método de entrada

La mayoría de los métodos de entrada de software proporcionan un botón de acción del usuario en la esquina inferior que es apropiado para el campo de texto actual. De manera predeterminada, el sistema usa este botón para la acción **Siguiente** o **Listo**, a menos que tu campo de texto permita texto de varias líneas (como con android:inputType="textMultiLine"), en cuyo caso el botón de acción es un retorno de carro. Sin embargo, puedes especificar acciones adicionales que podrían ser más apropiadas para tu campo de texto, como **Enviar** o **Ir**.

Para especificar el botón de acción del teclado, usa el atributo [android:imeOptions](https://developer.android.com/reference/android/widget/TextView.html#attr_android:imeOptions) con un valor de acción como "actionSend" o "actionSearch". Por ejemplo:



**Figura 4:** El botón Enviar aparece cuando declaras android:imeOptions="actionSend".

    <EditText  
        android:id="@+id/search"  
        android:layout\_width="fill\_parent"  
        android:layout\_height="wrap\_content"  
        android:hint="@string/search\_hint"  
        android:inputType="text"  
        android:imeOptions="actionSend" />

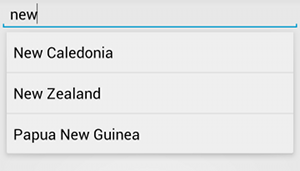
Luego, puedes escuchar las pulsaciones en el botón de acción si defines un [TextView.OnEditorActionListener](https://developer.android.com/reference/android/widget/TextView.OnEditorActionListener.html) para el elemento [EditText](https://developer.android.com/reference/android/widget/EditText.html). En tu objeto de escucha, responde al ID de acción IME apropiado definido en la clase [EditorInfo](https://developer.android.com/reference/android/view/inputmethod/EditorInfo.html), como [IME\_ACTION\_SEND](https://developer.android.com/reference/android/view/inputmethod/EditorInfo.html#IME_ACTION_SEND). Por ejemplo:

[KOTLIN](https://developer.android.com/training/keyboard-input/style.html#kotlin)[JAVA](https://developer.android.com/training/keyboard-input/style.html#java)

    EditText editText = (EditText) findViewById(R.id.search);  
    editText.setOnEditorActionListener(new OnEditorActionListener() {  
        @Override  
        public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {  
            boolean handled = false;  
            if (actionId == EditorInfo.IME\_ACTION\_SEND) {  
                sendMessage();  
                handled = true;  
            }  
            return handled;  
        }  
    });

Cómo proporcionar sugerencias de autocompletar

Si quieres proporcionar sugerencias a los usuarios mientras escriben, puedes usar una subclase de [EditText](https://developer.android.com/reference/android/widget/EditText.html) llamada [AutoCompleteTextView](https://developer.android.com/reference/android/widget/AutoCompleteTextView.html). Para implementar el autocompletado, debes especificar un elemento [Adapter](https://developer.android.com/reference/android/widget/Adapter.html) que proporcione las sugerencias de texto. Hay varios tipos de adaptadores disponibles, según el origen de los datos, como una base de datos o un arreglo.



**Figura 5:** Ejemplo de [AutoCompleteTextView](https://developer.android.com/reference/android/widget/AutoCompleteTextView.html) con sugerencias de texto

En el siguiente procedimiento, se describe cómo configurar un [AutoCompleteTextView](https://developer.android.com/reference/android/widget/AutoCompleteTextView.html) que proporciona sugerencias de un arreglo mediante [ArrayAdapter](https://developer.android.com/reference/android/widget/ArrayAdapter.html):

1. Agrega el elemento [AutoCompleteTextView](https://developer.android.com/reference/android/widget/AutoCompleteTextView.html) a tu diseño. El siguiente es un diseño con solo el campo de texto:

    <?xml version="1.0" encoding="utf-8"?>  
    <AutoCompleteTextView xmlns:android="http://schemas.android.com/apk/res/android"  
        android:id="@+id/autocomplete\_country"  
        android:layout\_width="fill\_parent"  
        android:layout\_height="wrap\_content" />

1. Define el arreglo que contiene todas las sugerencias de texto. Por ejemplo, aquí hay un arreglo de nombres de países que se define en un archivo de recursos XML (res/values/strings.xml):

    <?xml version="1.0" encoding="utf-8"?>  
    <resources>  
        <string-array name="countries\_array">  
            <item>Afghanistan</item>  
            <item>Albania</item>  
            <item>Algeria</item>  
            <item>American Samoa</item>  
            <item>Andorra</item>  
            <item>Angola</item>  
            <item>Anguilla</item>  
            <item>Antarctica</item>  
            ...  
        </string-array>  
    </resources>

1. En tu [Activity](https://developer.android.com/reference/android/app/Activity.html) o [Fragment](https://developer.android.com/reference/android/app/Fragment.html), usa el siguiente código para especificar el adaptador que proporciona las sugerencias:

[KOTLIN](https://developer.android.com/training/keyboard-input/style.html#kotlin)[JAVA](https://developer.android.com/training/keyboard-input/style.html#java)

    // Get a reference to the AutoCompleteTextView in the layout  
    AutoCompleteTextView textView = (AutoCompleteTextView) findViewById(R.id.autocomplete\_country);  
    // Get the string array  
    String[] countries = getResources().getStringArray(R.array.countries\_array);  
    // Create the adapter and set it to the AutoCompleteTextView  
    ArrayAdapter<String> adapter =  
            new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_1, countries);  
    textView.setAdapter(adapter);

Aquí, se inicializa un nuevo elemento [ArrayAdapter](https://developer.android.com/reference/android/widget/ArrayAdapter.html) para vincular cada elemento del arreglo de strings countries\_array a un objeto [TextView](https://developer.android.com/reference/android/widget/TextView.html) que existe en el diseño simple\_list\_item\_1 (este es un diseño proporcionado por Android que ofrece una apariencia estándar para el texto en una lista).

1. Asigna el adaptador a [AutoCompleteTextView](https://developer.android.com/reference/android/widget/AutoCompleteTextView.html) llamando a [setAdapter()](https://developer.android.com/reference/android/widget/AutoCompleteTextView.html#setAdapter(T)).