Si bien XML en sí mismo no es un lenguaje específico para diseñar interfaces, se utiliza comúnmente en conjunto con otras tecnologías, especialmente en el desarrollo de Android, para definir la estructura y el diseño de las interfaces de usuario. En este contexto, las etiquetas XML representan elementos de la interfaz, como botones, campos de texto, diseños, etc.

A continuación, te presento algunas de las etiquetas XML más utilizadas en el diseño de interfaces, especialmente en el contexto de Android:

Diseños (Layouts)

* `<LinearLayout>`: Organiza los elementos en una sola dirección, ya sea vertical u horizontal.
* `<RelativeLayout>`: Permite posicionar los elementos en relación con otros elementos o con el diseño principal.
* `<ConstraintLayout>`: Ofrece un sistema de diseño flexible y potente que permite crear interfaces complejas y adaptables a diferentes tamaños de pantalla.
* `<FrameLayout>`: Un contenedor simple que muestra un solo elemento a la vez.
* `<GridLayout>`: Organiza los elementos en una cuadrícula.

Vistas (Views)

* `<TextView>`: Muestra texto en la pantalla.
* `<EditText>`: Permite al usuario ingresar texto.
* `<Button>`: Un botón que el usuario puede tocar para realizar una acción.
* `<ImageView>`: Muestra imágenes.
* `<CheckBox>`: Una casilla de verificación que el usuario puede marcar o desmarcar.

* `<RadioButton>`: Un botón de opción que forma parte de un grupo, donde solo se puede seleccionar una opción.

* `<ProgressBar>`: Muestra una barra de progreso que indica una operación en curso.

* `<SeekBar>`: Permite al usuario seleccionar un valor dentro de un rango.

* `<ListView>`: Muestra una lista de elementos que se pueden desplazar.

* `<RecyclerView>`: Una versión más avanzada y eficiente de ListView.

Otros elementos

* `<include>`: Permite reutilizar diseños XML en diferentes partes de la interfaz.

* `<merge>`: Permite combinar diseños XML de manera eficiente.

* `<view>`: Un elemento genérico que se puede utilizar para representar cualquier tipo de vista.

Atributos comunes

* `android:id`: Identificador único del elemento.

* `android:layout_width`: Ancho del elemento.

* `android:layout_height`: Alto del elemento.

* `android:layout_margin`: Margen alrededor del elemento.

* `android:padding`: Espacio interior dentro del elemento.

* `android:visibility`: Controla si el elemento está visible, invisible u oculto.

* `android:background`: Define el fondo del elemento.

* `android:textColor`: Define el color del texto.

* `android:textSize`: Define el tamaño del texto.

Ejemplo

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

  <TextView
    android:id="@+id/nombre_usuario"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nombre del usuario" />

  <Button
    android:id="@+id/boton_guardar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Guardar" />

</LinearLayout>
```

Este es solo un ejemplo básico. Las posibilidades son muchas y la combinación de estas etiquetas y atributos te permitirá crear interfaces de usuario complejas y personalizadas.

Recursos adicionales

* **Documentación de Android Developers**: https://developer.android.com/

* **Ejemplos de diseños en Android**: https://developer.android.com/codelabs/basic-android-kotlin-training-xml-layouts?hl=es-419

Espero que esta información te sea útil. Si tienes alguna otra pregunta, no dudes en preguntar.