

# AWAKELAB

## #programmingbootcamp

# INTERFAZ GRÁFICA







## ¿QUÉ VAMOS A VER?

- ConstraintLayout
- Manejo de Strings
- Manejo de Colors
- Guidelines





**¡VAMOS A COMENZAR!**





# CONSTRAINT LAYOUT

# CONSTRAINT LAYOUT

- La principal característica de este tipo de layout es que sus objetos siempre deben estar ligados entre si y al contenedor en general.
- No debemos tener nunca un objeto que no interactúe con la estructura, salvo que su construcción base sea diferente.
- **parent:** Palabra reservada que hace referencia al layout.
- **Top\_toTopOf:** Especifica a que posición va a estar conectada la porción superior del objeto.





# CONSTRAINT LAYOUT

- **Bottom\_toBottomOf** también es una opción válida, al igual que **Start\_toStartOf** y **End\_toEndOf**.
- Una vez entendemos como funcionan las posiciones, podemos incluso combinarlas para conectarnos a otros objetos, por ejemplo **Top\_toBottomOf** quiere decir que enlazamos la parte superior de un objeto, con la inferior de otro.
- También tenemos **layout\_constraintHorizontal\_bias** que nos sirven para manipular con porcentajes la posición en que queremos mover nuestro objeto. Existe **Vertical\_bias** que funciona de la misma manera. (*Va de 0 a 1*).



# CONSTRAINT LAYOUT

- **<EditText />**: Nos va a permitir crear un campo de texto en el que el usuario puede ingresar diferentes datos.
- **<Button />**: Genera un botón con cierto diseño ya predefinido por Android Studio.
- **android:id**: Este atributo nos permitirá agregar un identificador único a cada objeto que tengamos en nuestro diseño. Dentro es importante utilizar la siguiente sintaxis ***"@+id/nombreID"***
- **android:hint**: Nos permite generar un texto a modo de instrucción para el usuario. En otros lenguajes de programación es conocido como *placeholder*.





# MANEJO DE STRINGS



# MANEJO DE STRINGS



- Lo que uno tiene a pensar es que cada título, nombre de botón o incluso, instrucción para el usuario, son textos aislados que podemos repartir a lo largo de nuestro código.
- **res > values > strings.xml**
- Dentro de este documento vamos a poder crear nuestros diferentes Strings a usar a lo largo de la aplicación.
- De esta manera, en vez de escribir un texto “forzado” dentro de la app, simplemente haremos referencia a su **name** y lo podremos tener en dónde nosotros queramos.





# MANEJO DE COLORES



# MANEJO DE COLORES

- Ya vimos que tenemos la oportunidad de usar colores con código hexadecimal directamente en el objeto, pero esto no es para nada una buena práctica.
- **res > values > colors.xml**
- Es por eso que debemos ir directamente a colors.xml para los colores propios de nuestra aplicación.
- Se recomienda el uso de <https://www.materialpalette.com> para escoger una paleta de colores.



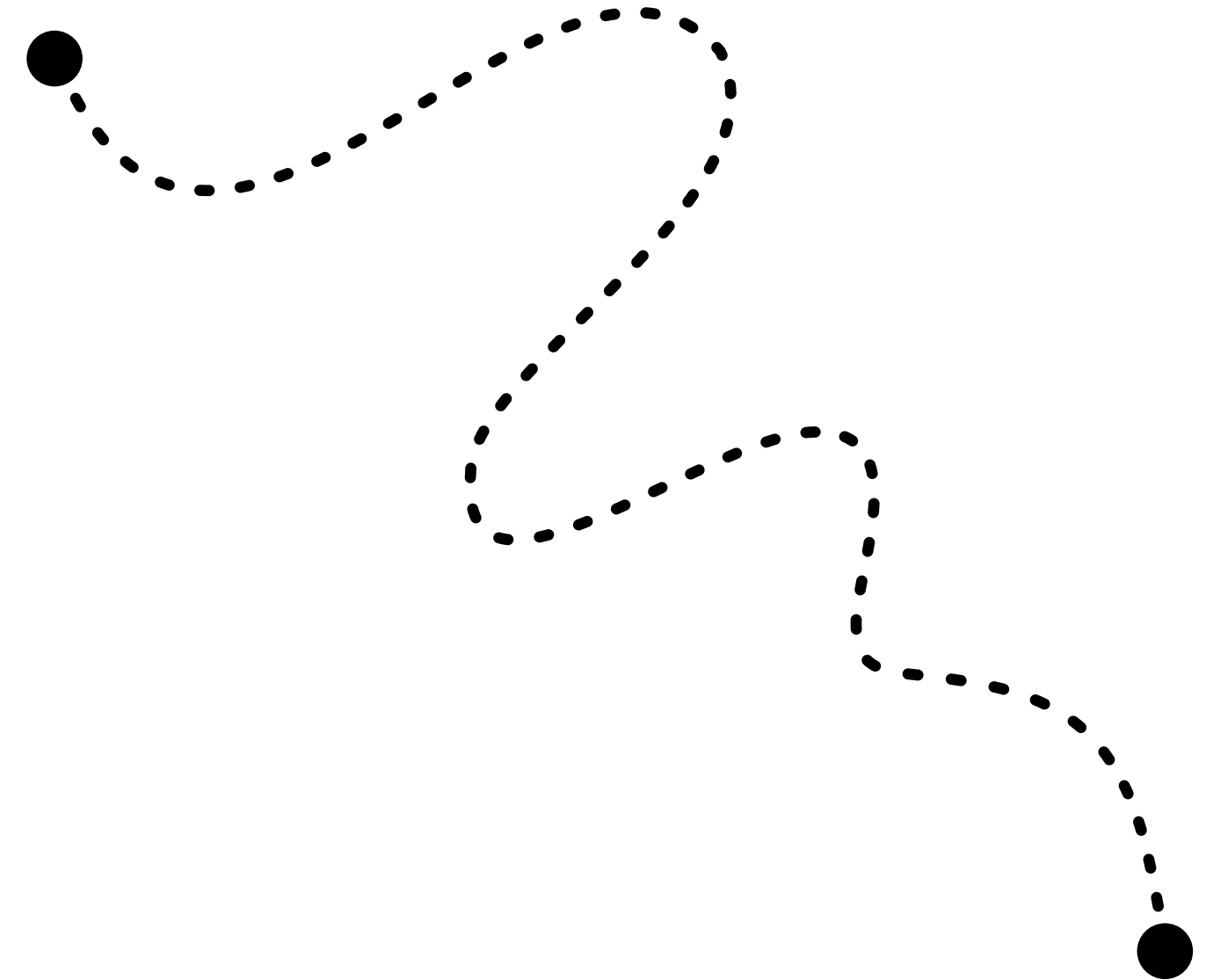




# GUIDELINES

# GUIDELINES

- Ya vimos que es posible enlazar nuestros objetos a cualquier estructura que tengamos dentro de nuestro layout, ya sea manualmente o por medio de código.
- Con los **guidelines** podemos crear líneas “invisibles” que nos van a permitir anclar los objetos a lugares que normalmente no podemos.
- Vamos a tener la facultad de utilizar la cantidad de líneas que nosotros queramos, además de que la orientación no tiene porque ser siempre la misma.



# GUIDELINES

- Más allá del sentido que le vamos a dar a nuestras líneas de guía, existen tres formas diferentes de usarla.
- La podemos anclar a la izquierda del todo de nuestro layout, es decir, **Start**.
- Por otro lado, también podemos hacer lo mismo pero a la derecha del todo, es decir, **End**.
- Aunque lo más recomendado es utilizar la tercera forma, es decir, por medio de porcentajes vamos a posicionar cada una de las líneas dentro del layout.
- **layout\_constraintGuide\_percent**







GRACIAS POR  
LA ATENCIÓN