Bootcamp - Android

Roshka - agosto 2022

Constraint layout

Un **ConstraintLayout** es un ViewGroup que permite colocar y dimensionar widgets de manera flexible.

Constraint es una conexión o alineamiento con otro elemento UI, al layout padre o a una guía invisible.

El ConstraintLayout es mejor cuando tienes que hacer vistas complicadas con una cantidad no muy grande de elementos.

Constraint layout Ventajas

- Podes hacer responsive a distintas pantallas y resoluciones
- Generalmente una jerarquía de view más plana
- Optimizada para poner distintos views
- Es fácil poner Views en cualquier lugar y el editor ayuda a colocar constraints

Tipos de constraint

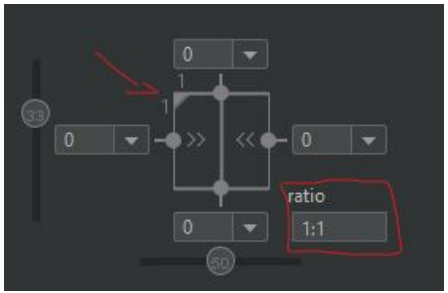
- Fix constraints representadas por una línea recgta, para dimensiones específicas
- Wrap content que es una linea de **>** las vistas se expanden lo suficiente como para cubrir su contenido
- Línea zigzag, que sería para matchear constraints, mientras más constraints existan, le será más fácil al layout acomodarse en distintas pantallas, lo que significa que necesitamos menos layouts para la app

Ratio

Podemos definir una dimensión de nuestros widgets como el **ratio** de nuestra otra dimensión.

Por ejemplo podemos definir un cuadrado con un ratio de 1:1. Para lograr esto primero definimos como constraint el alto o el ancho de nuestro widget y el otro con una dimensión de 0dp.

Luego podremos clicar la opción de ratio y definir la relación de tamaño entre uno y otro.



Chains

Los **chains** son constraints para encadenar views horizontal o verticalmente y manejarlas como un grupo.

Se considera chain a views que están conectadas bidireccionalmente.

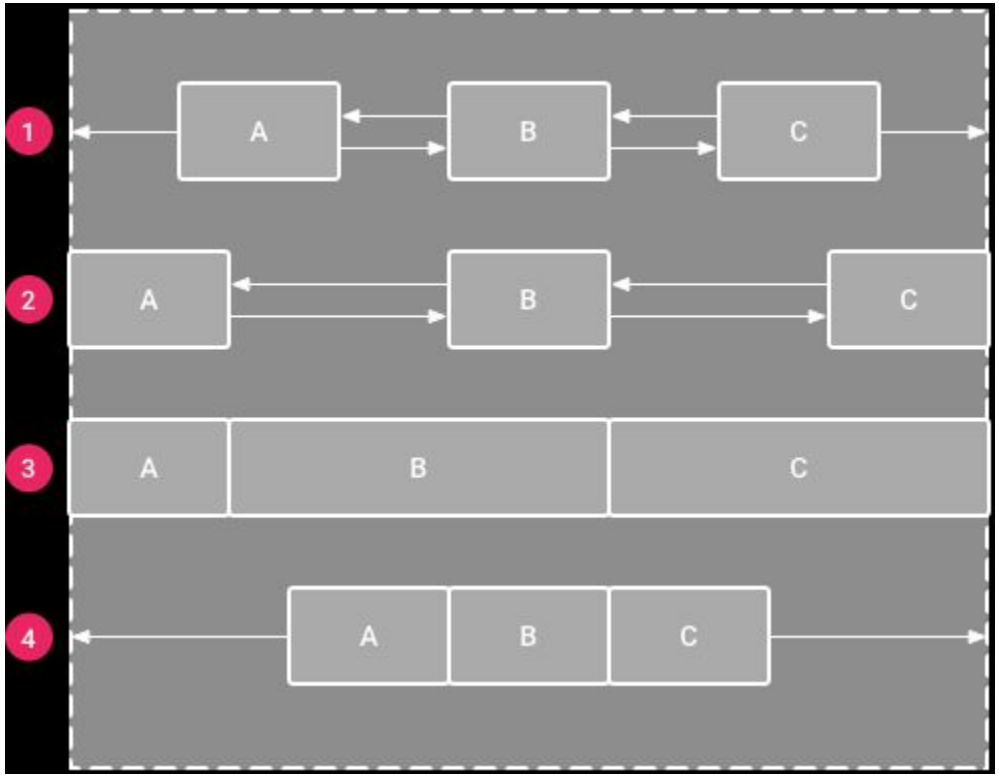
No se necesita nada más para que sea una cadena, solo conectar la cadena en ambas direcciones.

Chains

Se puede ordenar de 4 formas distintas:

1. **Spread**: las vistas se distribuyen uniformemente (después de tener en cuenta los márgenes). Este es el valor predeterminado.
2. **Spread inside**: La primera y la última vista se fijan a las restricciones en cada extremo de la cadena y el resto se distribuye uniformemente.
3. **Weighted**: Usará todo el espacio y redimensionará los elementos para encajar, basado en los valores del horizontal o vertical layout constraint.
4. **Packed**: las vistas se empaquetan juntas, lo contrario a weighted.

Chains



Baseline

Los **baselines** son constraint para alinear elementos en una línea base como su nombre lo dice.

Podemos usarlo cuando queremos alinear texto de diferentes tamaños o cuando queremos alinear distintos botones.

Tarea

Realizar una pantalla similar a la de la imagen para registrarse en un banco. Utilizar Constraint Layout

Debe tener:

- Campo fecha(puede ser DatePicker)
- Sexo(como lista)
- Estado civil (como lista)
- Numero de telefono(debe controlar que empiece con 09 y no debe permitir ingresar letras, ni copiando ni pegando)
- Un correo electronico (debe contener ser un correo valido contener @ y terminar en .algo)
- Debe ingresar su RUC (el ruc solo permite numeros, 1 guion y solo 1 numero al final ejemplo: **4648961-6**)