Cómo crear una IU receptiva con ConstraintLayout  **Parte de**[**Android Jetpack**](https://developer.android.com/jetpack)**.**

[ConstraintLayout](https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout.html) te permite crear diseños grandes y complejos con una jerarquía de vista plana (sin grupos de vistas anidadas). Es similar a [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout.html) en cuanto a que se presentan todas las vistas de acuerdo con las relaciones entre las vistas equivalentes y el diseño principal, pero es más flexible que RelativeLayout y más fácil de usar con el Editor de diseño de Android Studio.

Toda la potencia de ConstraintLayout está disponible directamente desde las herramientas visuales del Editor de diseño, ya que la API de diseño y el Editor de diseño se crearon de forma específica para funcionar en conjunto. De esta manera, puedes crear tu diseño con ConstraintLayout por completo arrastrando y soltando, en lugar de editando el XML.

En esta página, se incluye una guía para crear un diseño con [ConstraintLayout](https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout.html) en Android Studio 3.0 o versiones posteriores. Si deseas obtener más información sobre el Editor de diseño, consulta la guía de Android Studio sobre [Cómo crear una IU con el Editor de diseño](https://developer.android.com/studio/write/layout-editor.html).

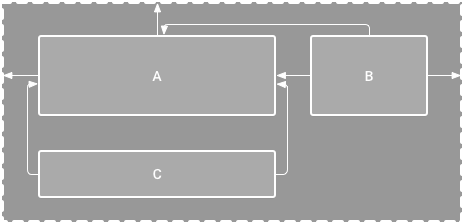
Para ver una serie de diseños que puedes crear con ConstraintLayout, consulta el [proyecto de ejemplos de Constraint Layout en GitHub](https://github.com/android/views-widgets-samples/tree/master/ConstraintLayoutExamples).

Descripción general de las restricciones

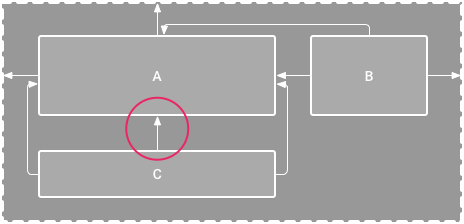
Para definir la posición de una vista en ConstraintLayout, debes agregar al menos una restricción horizontal y una vertical. Cada restricción representa una conexión o alineación con otra vista, el diseño superior o una guía invisible. Cada restricción define la posición de la vista a lo largo del eje vertical u horizontal, por lo que cada vista debe tener un mínimo de una restricción para cada eje, aunque a menudo se necesitan más.

Cuando sueltas una vista en el Editor de diseño, esta permanece donde la dejas, incluso si no tiene restricciones. Sin embargo, esto solo sirve para facilitar la edición. Si una vista no tiene restricciones cuando ejecutas su diseño en un dispositivo, se abre en la posición [0,0] (la esquina superior izquierda).

En la figura 1, el diseño se ve bien en el editor, pero no hay restricción vertical en la vista C. Cuando este diseño se abre en un dispositivo, la vista C se alinea horizontalmente con los bordes izquierdo y derecho de la vista A, pero aparece en la parte superior de la pantalla porque no tiene restricción vertical.



**Figura 1**: El editor muestra la vista C debajo de la A, pero no tiene restricción vertical



**Figura 2**: La vista C ahora tiene una restricción vertical debajo de la vista A

Si bien la falta de una restricción no causará un error de compilación, el Editor de diseño mostrará un error en la barra de herramientas cuando falten restricciones. Para ver los errores y otras advertencias, haz clic en **Show Warnings and Errors** . Para evitar que falten restricciones, el Editor de diseño puede agregar restricciones automáticamente con las funciones [Infer Constraints y Autoconnect](https://developer.android.com/training/constraint-layout/index.html#use-autoconnect-and-infer-constraints).

Cómo agregar ConstraintLayout a tu proyecto

Para usar ConstraintLayout en tu proyecto, Sigue estos pasos:

1. Asegúrate de tener el repositorio maven.google.com declarado en tu archivo build.gradle de nivel de módulo:

    repositories {  
        google()  
    }

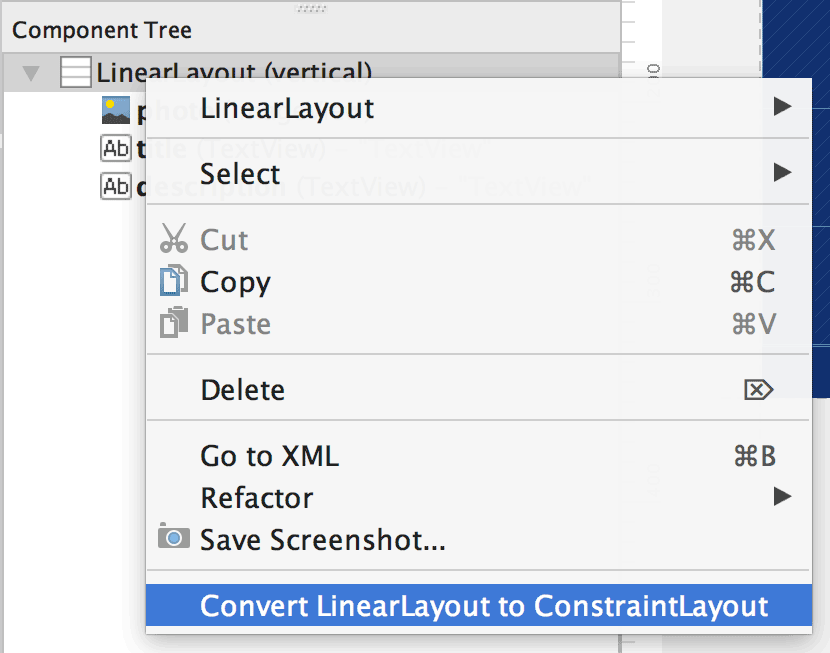
1. Agrega la biblioteca como una dependencia en el mismo archivo build.gradle, como se muestra en el siguiente ejemplo. Ten en cuenta que la última versión puede ser diferente a la que se muestra en el ejemplo:

    dependencies {  
        implementation 'com.android.support.constraint:constraint-layout:1.1.2'  
    }

1. En la barra de herramientas o notificación de sincronización, haz clic en **Sync Project with Gradle Files**.

Ya tienes todo listo para crear tu diseño con ConstraintLayout.

Cómo convertir un diseño



**Figura 3**: Menú para convertir un diseño en ConstraintLayout

Para convertir un diseño existente en un diseño de restricción, sigue estos pasos:

1. Abre tu diseño existente en Android Studio y haz clic en la pestaña **Design**, en la parte inferior de la ventana del editor.
2. En la ventana **Component Tree**, haz clic con el botón derecho en el diseño y, luego, en Convert *layout* to ConstraintLayout.

Cómo crear un nuevo diseño

Para iniciar un nuevo archivo de diseño de restricciones, sigue estos pasos:

1. En la ventana **Project**, haz clic en la carpeta del módulo y, luego, selecciona File > New > XML > Layout XML.
2. Ingresa un nombre para el archivo de diseño e ingresa "android.support.constraint.ConstraintLayout" en **Root Tag**.
3. Haz clic en **Finish**.

Cómo agregar o quitar una restricción

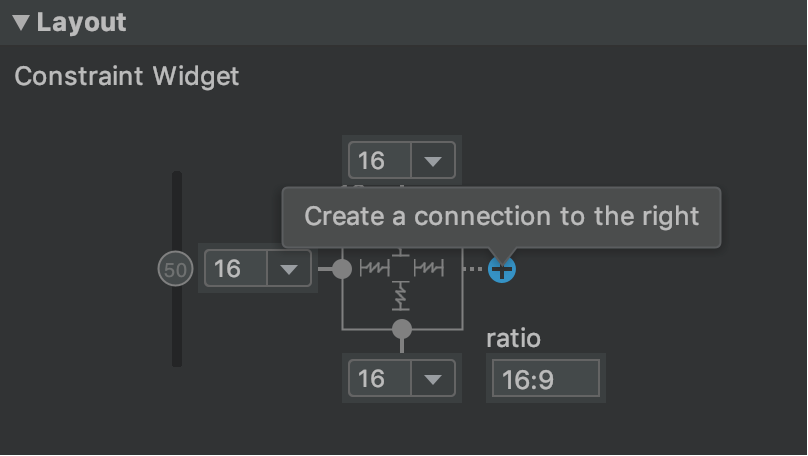
Para agregar una restricción, haz lo siguiente:

**Video 1**: El lado izquierdo de una vista está restringido al lado izquierdo del elemento superior

1. Arrastra una vista desde la ventana **Palette** hasta el editor.

Cuando agregas una vista en un ConstraintLayout, se muestra un cuadro delimitador con controladores de cambio de tamaño (cuadrados) en cada esquina y controladores de restricciones (círculos) en cada lado.

1. Haz clic en la vista para seleccionarla.
2. Realiza alguna de las siguientes acciones:
   * Haz clic en un controlador de restricción y arrástralo hasta un punto de anclaje disponible. Este punto puede ser el borde de otra vista, el borde del diseño o una guía. Ten en cuenta que, a medida que arrastras el controlador de restricciones, el editor de diseño muestra los posibles anclajes de conexión y las superposiciones azules.
   * Haz clic en uno de los botones **Create a connection**  en la sección Layout de la ventana Attributes, como se muestra en la figura 4.



**Figura 4**: La sección **Layout** de la ventana **Attributes** te permite crear conexiones.

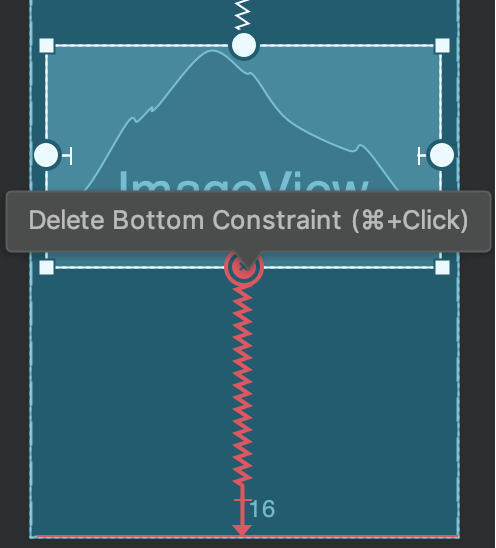
Cuando se crea la restricción, el editor le otorga un [margen predeterminado](https://developer.android.com/training/constraint-layout/index.html#adjust-the-view-margins) para separar las dos vistas.

Cuando crees restricciones, recuerda las siguientes reglas:

* Cada vista debe tener al menos dos restricciones, una horizontal y una vertical.
* Puedes crear restricciones solo entre un controlador de restricción y un punto de anclaje que compartan el mismo plano. Por lo tanto, un plano vertical (los lados izquierdo y derecho) de una vista puede limitarse solo a otro plano vertical; y las líneas de base solo pueden limitarse a otras líneas de base.
* Cada identificador de restricción se puede usar para una sola restricción, pero puedes crear múltiples restricciones (desde diferentes vistas) en el mismo punto de anclaje.

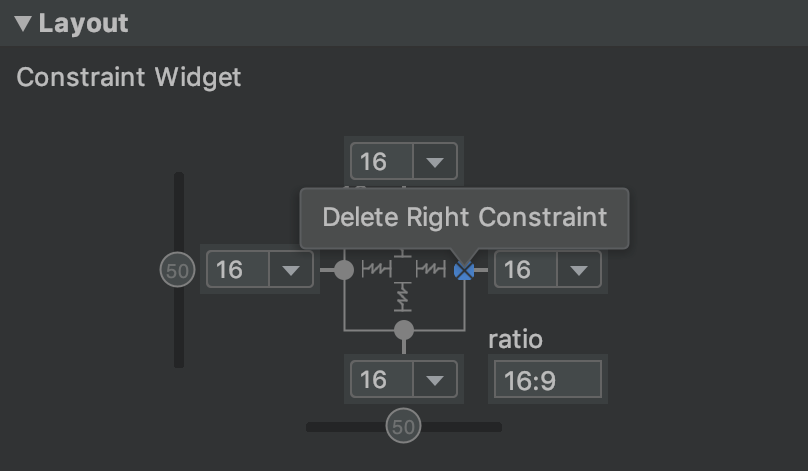
Para borrar una restricción, haz lo siguiente:

* Haz clic en una restricción para seleccionarla y, luego, presiona Delete.
* Mantén presionado Control (Command en macOS) y, luego, haz clic en un anclaje de restricción. Ten en cuenta que, cuando la restricción se vuelve de color rojo, esto indica que puedes hacer clic para borrarla, como se muestra en la figura 5.



**Figura 5**: Una restricción roja indica que puedes hacer clic para borrarla

* En la sección **Layout** de la ventana Attributes, haz clic en un anclaje de restricción, como se muestra en la figura 6.



**Figura 6**: Haz clic en un anclaje de restricciones para borrarlo

**Video 2**: Cómo agregar una restricción opuesta a una existente

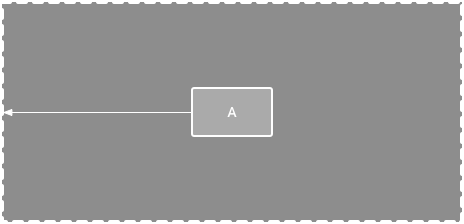
Si agregas restricciones opuestas en una vista, las líneas de restricción se vuelven onduladas como un resorte para indicar las fuerzas opuestas, como se muestra en el video 2. El efecto es más visible cuando el tamaño de la vista se define como "Fixed" o "Wrap content", en cuyo caso la vista se centra entre las restricciones. Si, en cambio, deseas que la vista se expanda para cumplir con las restricciones, [cambia el tamaño a "Match constraints"](https://developer.android.com/training/constraint-layout/index.html#adjust-the-view-size); o si deseas mantener el tamaño actual, pero mover la vista para que no esté centrada, [ajusta el sesgo de restricciones](https://developer.android.com/training/constraint-layout/index.html#adjust-the-constraint-bias).

Puedes usar restricciones para lograr diferentes tipos de comportamiento de diseño, como se describe en las siguientes secciones.

Posición superior

Restringe el lado de una vista al borde correspondiente del diseño.

En la figura 7, el lado izquierdo de la vista está conectado al borde izquierdo del diseño superior. Puedes definir la distancia desde el borde con el margen.

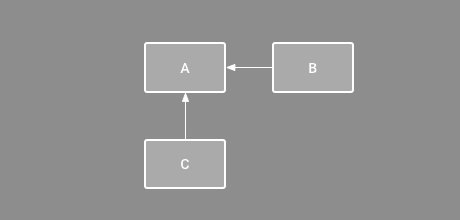


**Figura 7**: Una restricción horizontal para el elemento superior

Posición de orden

Define el orden de aparición de dos vistas, ya sea de manera horizontal o vertical.

En la figura 8, la vista B está restringida a estar siempre a la derecha de A, y C está restringida debajo de A. Sin embargo, estas restricciones no implican alineación, por lo que B puede moverse hacia arriba y hacia abajo.



**Figura 8**: Una restricción horizontal y vertical

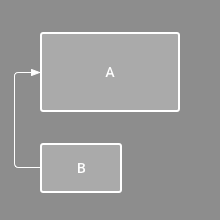
Alineación

Alinea el borde de una vista con el mismo borde de otra vista.

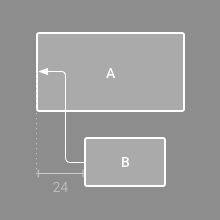
En la figura 9, el lado izquierdo de B está alineado con el lado izquierdo de A. Si deseas alinear los centros de las vistas, crea una restricción en ambos lados.

Puedes compensar la alineación arrastrando la vista desde la restricción hacia adentro. Por ejemplo, en la figura 10, se muestra la vista B con una alineación de desplazamiento de 24dp. El desplazamiento está definido por el margen de la vista restringida.

También puedes seleccionar todas las vistas que deseas alinear y, luego, hacer clic en **Align**  en la barra de herramientas para seleccionar el tipo de alineación.



**Figura 9**: Una restricción de alineación horizontal



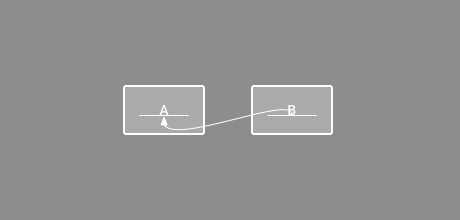
**Figura 10**: Una restricción de alineación horizontal desplazada

Alineación de línea de base

Alinea la línea de base de texto de una vista con la línea de base de texto de otra vista.

En la figura 11, la primera línea de B está alineada con el texto de A.

Para crear una restricción de línea de base, haz clic con el botón derecho en la vista de texto que deseas restringir y, luego, haz clic en **Show Baseline**. A continuación, haz clic en la línea de base de texto y arrastra la línea a otra línea de base.



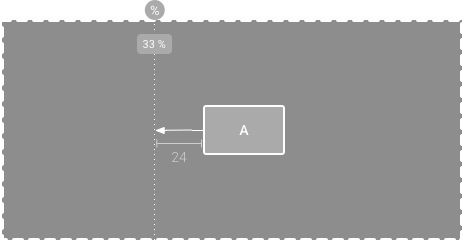
**Figura 11**: Una restricción de alineación de línea de base

Cómo aplicar una restricción a una guía

Puedes agregar una guía vertical u horizontal en la que puedes restringir las vistas, y la guía será invisible para los usuarios de la aplicación. Puedes colocar la guía dentro del diseño según las unidades de dp o porcentaje, en relación con el borde del diseño.

Para crear una guía, haz clic en **Guidelines**  en la barra de herramientas y, luego, haz clic en Add Vertical Guideline o Add Horizontal Guideline.

Arrastra la línea punteada para cambiar su posición y haz clic en el círculo que se encuentra en el borde de la guía para activar o desactivar el modo de medición.



**Figura 12**: Una vista restringida a una guía

Cómo aplicar una restricción a una barrera

Del mismo modo que las guías, una barrera es una línea invisible en la que puedes restringir vistas. Excepto cuando una barrera no defina su propia posición, esta se moverá en función de la posición de las vistas contenidas en ella. Esto es útil si deseas restringir una vista al conjunto de vistas en lugar de a una vista específica.

Por ejemplo, en la figura 13, se muestra que la vista C está limitada al lado derecho de una barrera. La barrera se establece en el "extremo" (o el lado derecho en un diseño de izquierda a derecha) de la vista A y la vista B. De este modo, el movimiento de la barrera depende de si el lado derecho de la vista A o la vista B es el que está más a la derecha.

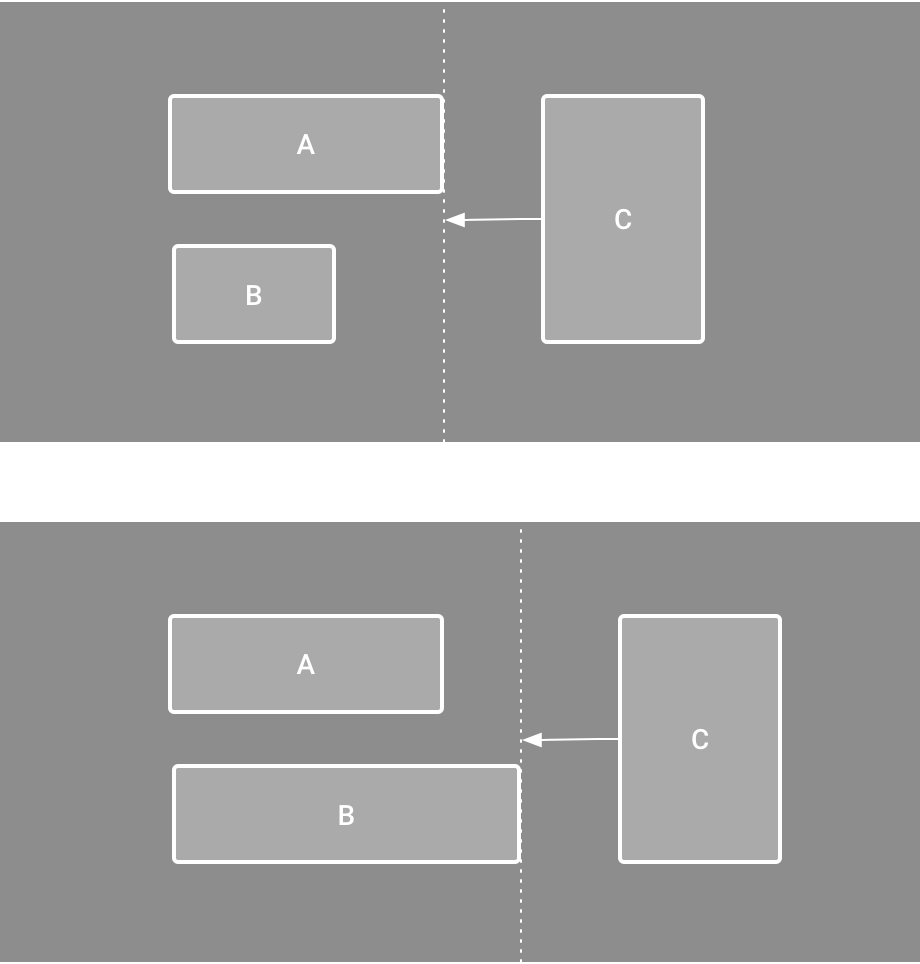
Para crear una barrera, sigue estos pasos:

1. Haz clic en **Guidelines**  en la barra de herramientas y, luego, en Add Vertical Barrier o Add Horizontal Barrier.
2. En la ventana **Component Tree**, selecciona las vistas que deseas dentro de la barrera y arrástralas hasta el componente de barrera.
3. Selecciona la barrera en **Component Tree**, abre la ventana Attributes  y, luego, configura la barrierDirection.

Ahora puedes crear una restricción desde otra vista hasta la barrera.

También puedes restringir a la barrera las vistas que están *dentro* de la barrera. De esta manera, puedes asegurarte de que todas las vistas de la barrera siempre estén alineadas entre sí, aunque no sepas cuál de las vistas será la más larga o la más alta.

También puedes incluir una guía dentro de una barrera para garantizar una posición "mínima" para la barrera.



**Figura 13**: La vista C está restringida a una barrera, que se mueve según la posición o el tamaño de las vistas A y B

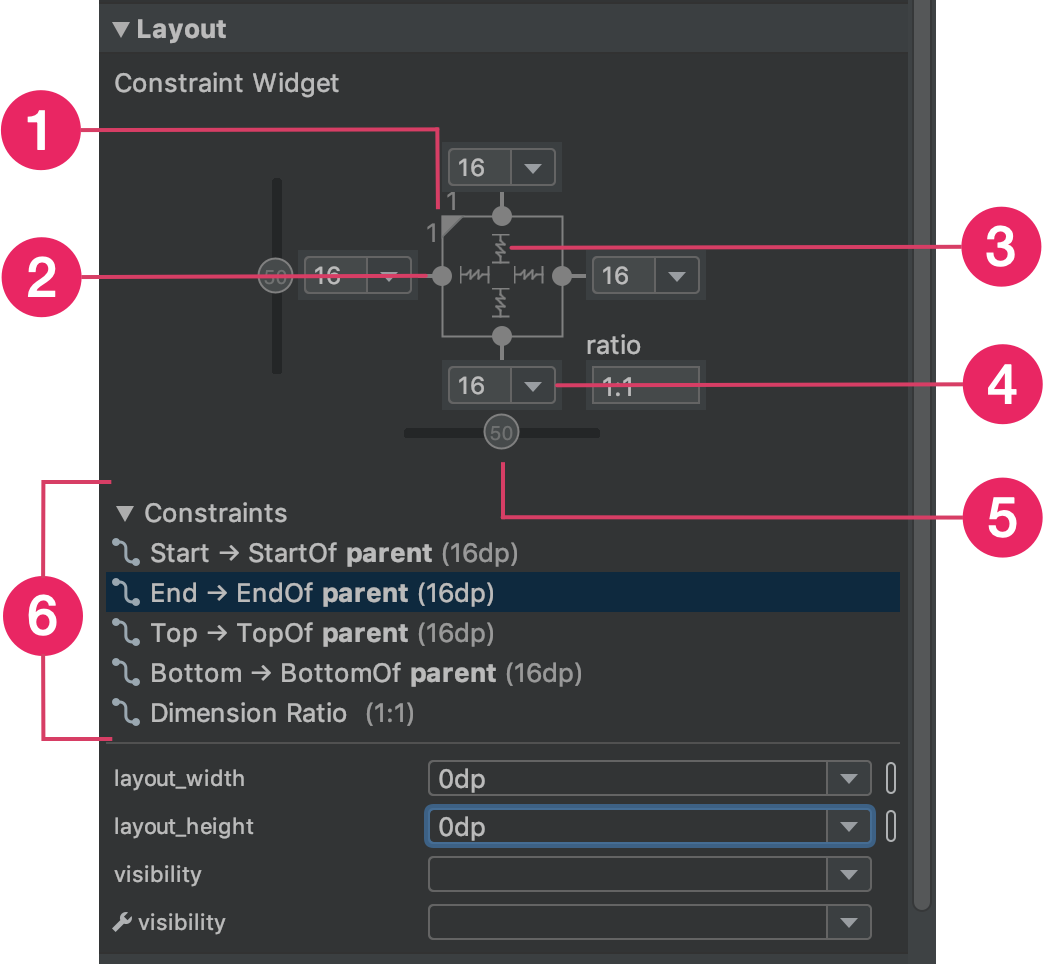
Cómo ajustar el sesgo de restricciones

Cuando agregas una restricción a ambos lados de una vista (y el tamaño de la misma dimensión es "Fixed" o "Wrap content"), la vista se centra entre las dos restricciones con un sesgo del 50% de forma predeterminada. Para ajustar el sesgo, arrastra el control deslizante en la ventana **Attributes** o arrastra la vista, como se muestra en el video 3.

En cambio, si deseas que la vista se estire para cumplir con las restricciones, [cambia el tamaño para "ajustar las restricciones"](https://developer.android.com/training/constraint-layout/index.html#adjust-the-view-size).

**Video 3**: Cómo ajustar el sesgo de restricciones

Cómo ajustar el tamaño de la vista



**Figura 14**: Cuando seleccionas una vista, la ventana **Attributes** incluye controles para 1 relación de tamaño, 2 borrar restricciones, 3 modo de altura/ancho, 4 márgenes y 5 sesgo de restricciones. También puedes destacar restricciones individuales en el Editor de diseño haciendo clic en ellas en la 6 lista de restricciones.

Puedes usar los controladores de las esquinas para cambiar el tamaño de una vista, pero esto establece el tamaño dentro del código a fin de que la vista no cambie de tamaño para diferentes contenidos o tamaños de pantalla. Si quieres seleccionar un modo de tamaño diferente, haz clic en una vista y abra la ventana **Attributes**  en el lado derecho del editor.

Cerca de la parte superior de la ventana **Attributes**, se encuentra el inspector de vistas, que incluye controles para varios atributos de diseño, como se muestra en la figura 14 (disponible solo para vistas en un diseño de restricciones).

Para cambiar la forma en que se calculan la altura y el ancho, haz clic en los símbolos que se indican con la leyenda 3 en la figura 14. Estos símbolos representan el modo de tamaño de la siguiente manera (haz clic en el símbolo para alternar entre los diferentes parámetros de configuración):

* **Fixed**: Debes definir una dimensión específica en el cuadro de texto a continuación o cambiar el tamaño de la vista en el editor.
* **Wrap Content**: La vista se expande solo lo necesario para ajustarse a su contenido.
* **Match Constraints**: La vista se expande tanto como sea posible para cumplir con las restricciones de cada lado (después de tener en cuenta los márgenes de la vista). Sin embargo, puedes modificar ese comportamiento con los siguientes atributos y valores (estos atributos solo tienen efecto cuando estableces el ancho de la vista para que coincida con las restricciones):
  + **layout\_constraintWidth\_default**
    - **spread**: Expande la vista tanto como sea posible para cumplir con las restricciones de cada lado. Este es el comportamiento predeterminado.
    - **wrap**: Expande la vista solo lo necesario para ajustar su contenido, pero permite que la vista sea más pequeña si las restricciones lo requieren. Por lo tanto, la diferencia entre esto y usar **Wrap Content** (arriba) es que cuando se establece el ancho en **Wrap Content**, el ancho siempre debe coincidir exactamente con el ancho del contenido; mientras que cuando se usa **Match Constraints** con **layout\_constraintWidth\_default** establecido en **wrap**, también se permite que la vista sea más pequeña que el ancho del contenido.
  + **layout\_constraintWidth\_min**

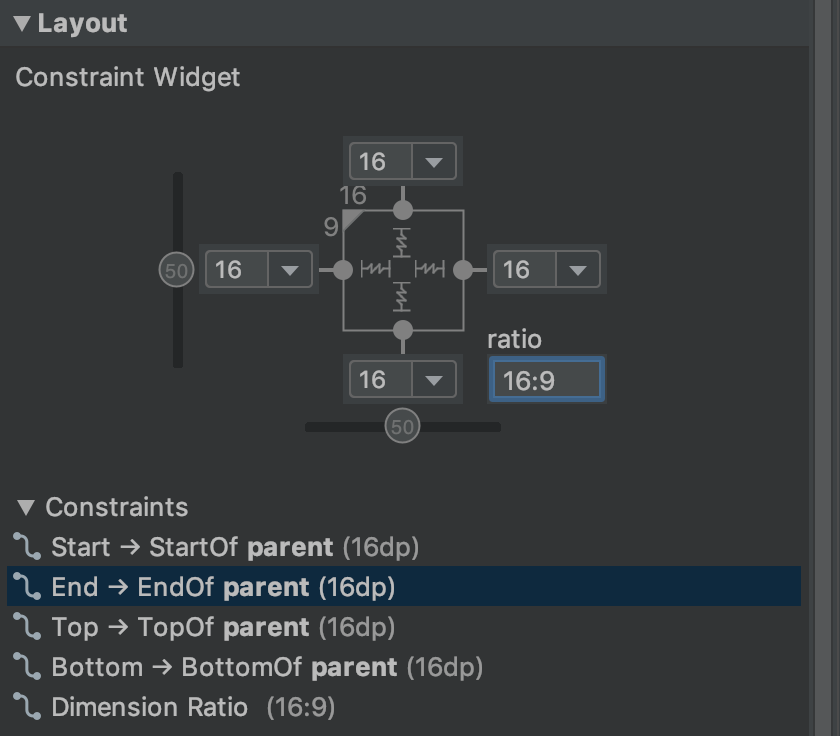
Toma una dimensión dp para el ancho mínimo de la vista.

* + **layout\_constraintWidth\_max**

Toma una dimensión dp para el ancho máximo de la vista.

* Sin embargo, si la dimensión determinada tiene una sola restricción, la vista se expande para ajustarse a su contenido. El uso de este modo en el ancho o la altura también te permite [establecer una relación de tamaño](https://developer.android.com/training/constraint-layout/index.html#ratio).

**Nota**: No puedes usar **match\_parent** para ninguna vista en un **ConstraintLayout**. En cambio, debes usar "Match Constraints" (**0dp**).



**Figura 15**: La vista se establece con un aspecto de 16:9, y el ancho se basa en una relación de la altura.

Cómo establecer el tamaño como una relación

Puedes establecer el tamaño de la vista en una relación, como 16:9, si al menos una de las dimensiones de la vista está configurada como "Match Constraints" (0dp). Para habilitar la relación, haz clic en **Toggle Aspect Ratio Constraint** (leyenda 1 en la figura 13) y, luego, ingresa la relación de ***ancho***:***altura*** en la entrada que se muestra.

Si tanto el ancho como la altura están configurados para coincidir con las restricciones, puedes hacer clic en **Toggle Aspect Ratio Constraint** para seleccionar qué dimensión se basa en una relación de la otra. El inspector de vistas conecta los bordes correspondientes con una línea continua para indicar qué vista se establece como una relación.

Por ejemplo, si estableces ambos lados como "Match Constraints", haz clic en **Toggle Aspect Ratio Constraint** dos veces para definir el ancho como una relación de la altura. Ahora todo el tamaño depende de la altura de la vista (que se puede definir de cualquier forma), como se muestra en la figura 15.

Cómo ajustar los márgenes de la vista

Para asegurarte de que todas tus vistas estén espaciadas de manera uniforme, haz clic en **Margin**  en la barra de herramientas para seleccionar el margen predeterminado para cada vista que agregas al diseño. Cualquier cambio que realices en el margen predeterminado se aplicará solo a las vistas que agregues a partir de ese momento.

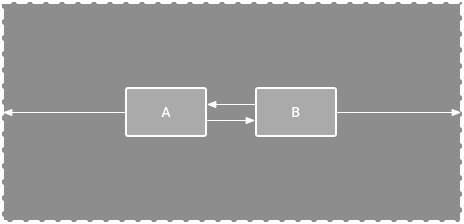
Puedes controlar el margen de cada vista en la ventana **Attributes** haciendo clic en el número de la línea que representa cada restricción (en la figura 13, la leyenda 4 muestra que se estableció el margen inferior en 16dp).



**Figura 16**: El botón **Margen** de la barra de herramientas

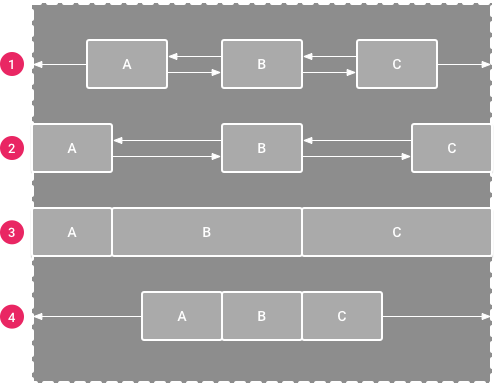
Todos los márgenes que ofrece la herramienta son factores de 8pb, lo que te ayudará a alinear tus vistas con las [recomendaciones de cuadrícula cuadrada de 8dp](https://material.google.com/layout/metrics-keylines.html) de Material Design.

Control de grupos lineales con una cadena



**Figura 17**: Una cadena horizontal con dos vistas

Una cadena es un grupo de vistas que están vinculadas entre sí con restricciones de posición bidireccionales. Dentro de una cadena, se pueden distribuir las vistas de manera horizontal o vertical.



**Figura 18**: Ejemplos de cada estilo de cadena

Es posible diseñar las cadenas de una de las siguientes maneras:

1. **Spread**: Las vistas se distribuyen uniformemente (después de que se tienen en cuenta los márgenes). Es el valor predeterminado.
2. **Spread inside**: La primera y la última vista se fijan a las restricciones de cada extremo de la cadena y el resto se distribuyen de manera uniforme.
3. **Weighted**: Cuando se define la cadena como spread o spread inside, puedes llenar el espacio restante configurando una o más vistas como "Match Constraints" (0dp). De manera predeterminada, se distribuye el espacio de uniformemente entre cada vista configurada para hacer coincidir las restricciones, pero puedes asignar una ponderación de importancia a cada vista utilizando los atributos layout\_constraintHorizontal\_weight y layout\_constraintVertical\_weight. Si tienes conocimientos sobre el layout\_weight en un [diseño lineal](https://developer.android.com/guide/topics/ui/layout/linear.html), recuerda que esto funciona de la misma manera. Por lo tanto, la vista con el valor de ponderación más alto obtiene la mayor cantidad de espacio; las vistas que tengan la misma ponderación obtendrán la misma cantidad de espacio.
4. **Packed**: las vistas se agrupan (una vez que se tienen en cuenta los márgenes). Luego, puedes ajustar el sesgo de toda la cadena (izquierda/derecha o arriba/abajo) cambiando el sesgo de la vista de extremo de la cadena.

La vista de "extremo" de la cadena (la vista más a la izquierda en una cadena horizontal y la vista superior en una cadena vertical) define el estilo de la cadena en XML. Sin embargo, puedes alternar entre **spread**, **spread inside** y **packed** seleccionando cualquier vista en la cadena y haciendo clic en el botón de la cadena  que aparece debajo de la vista.

Para crear una cadena, selecciona todas las vistas que se incluirán en la cadena, haz clic con el botón derecho en una de las vistas, selecciona **Chains** y, luego, Center Horizontally o Center Vertically, como se muestra en el video 4:

**Video 4**: Cómo crear una cadena horizontal

A continuación, se mencionan otras cuestiones que debes tener en cuenta cuando usas cadenas:

* Una vista puede formar parte de una cadena horizontal y vertical, lo que facilita la creación de diseños de cuadrícula flexibles.
* Una cadena funciona correctamente solo si cada extremo de la cadena está restringido a otro objeto en el mismo eje, como se muestra en la figura 14.
* Aunque la orientación de una cadena es vertical u horizontal, cuando se usa una no se alinean las vistas en esa dirección. Por lo tanto, debes asegurarte de incluir otras restricciones para lograr la posición adecuada para cada vista en la cadena, como [restricciones de alineación](https://developer.android.com/training/constraint-layout/index.html#alignment).

Cómo crear restricciones de manera automática

En lugar de agregar restricciones a cada vista a medida que las colocas en el diseño, puedes mover cada vista a las posiciones que deseas y, luego, hacer clic en **Infer Constraints**  para crear restricciones de manera automática.

**Infer Constraints** escanea el diseño para determinar el conjunto de restricciones más efectivo para todas las vistas. Su tarea es restringir las vistas a sus posiciones actuales y, al mismo tiempo, permitir la flexibilidad. Es posible que debas realizar algunos ajustes para asegurarte de que el diseño responda como lo deseas ante distintos tamaños de pantalla y orientaciones.

**Autoconnect to parent** es una función separada que también puedes habilitar. Si habilitas esta opción, cuando agregues vistas secundarias a una superior, esta función creará automáticamente dos o más restricciones para cada vista a medida que las agregues al diseño, pero solo cuando es apropiado restringir la vista al diseño superior. La conexión automática no crea restricciones para otras vistas del diseño.

La conexión automática está inhabilitada de manera predeterminada. Para habilitarla, haz clic en **Enable Autoconnection to Parent**  en la barra de herramientas del Editor de diseño.

Animaciones de fotogramas clave

Dentro de un ConstraintLayout, puedes animar los cambios en el tamaño y la posición de los elementos utilizando [ConstraintSet](https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintSet) y [TransitionManager](https://developer.android.com/reference/android/transition/TransitionManager).

**Nota**: **TransitionManager** está disponible en la biblioteca de compatibilidad de Android 4.0 (API nivel 14) o posterior.

Un ConstraintSet es un objeto liviano que representa las restricciones, los márgenes y el padding de todos los elementos secundarios dentro de un ConstraintLayout. Cuando aplicas un ConstraintSet a un ConstraintLayout que se muestra, el diseño actualiza las restricciones de todos sus elementos secundarios.

Para compilar una animación usando ConstraintSets, especifica dos archivos de diseño que actúen como un fotograma clave de inicio y fin para la animación. Luego, puedes cargar un ConstraintSet del segundo archivo de fotogramas clave y aplicarlo al ConstraintLayout que se muestra.

**Importante**: Las animaciones de ConstraintSet animan solo el tamaño y la posición de los elementos secundarios. No animan otros atributos (como el color).

En el siguiente ejemplo de código, se muestra cómo realizar una animación moviendo un solo botón hasta la parte inferior de la pantalla.

// MainActivity.kt  
  
    fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.keyframe\_one)  
        constraintLayout = findViewById(R.id.constraint\_layout) // member variable  
    }  
  
    fun animateToKeyframeTwo() {  
        val constraintSet = ConstraintSet()  
        constraintSet.load(this, R.layout.keyframe\_two)  
        TransitionManager.beginDelayedTransition()  
        constraintSet.applyTo(constraintLayout)  
    }

    // layout/keyframe1.xml  
    // Keyframe 1 contains the starting position for all elements in the animation as well as final colors and text sizes  
  
    <?xml version="1.0" encoding="utf-8"?>  
    <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:app="http://schemas.android.com/apk/res-auto"  
        android:layout\_width="match\_parent"  
        android:layout\_height="match\_parent">  
  
        <Button  
            android:id="@+id/button2"  
            android:layout\_width="0dp"  
            android:layout\_height="wrap\_content"  
            android:text="Button"  
            app:layout\_constraintEnd\_toEndOf="parent"  
            app:layout\_constraintStart\_toStartOf="parent"  
            **app:layout\_constraintTop\_toTopOf="parent"** />  
    </android.support.constraint.ConstraintLayout>

    // layout/keyframe2.xml  
    // Keyframe 2 contains another ConstraintLayout with the final positions  
  
    <?xml version="1.0" encoding="utf-8"?>  
    <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:app="http://schemas.android.com/apk/res-auto"  
        android:layout\_width="match\_parent"  
        android:layout\_height="match\_parent">  
  
        <Button  
            android:id="@+id/button2"  
            android:layout\_width="0dp"  
            android:layout\_height="wrap\_content"  
            android:text="Button"  
            app:layout\_constraintEnd\_toEndOf="parent"  
            app:layout\_constraintStart\_toStartOf="parent"  
            **app:layout\_constraintBottom\_toBottomOf="parent"** />  
    </android.support.constraint.ConstraintLayout>

Recursos adicionales

ConstraintLayout se utiliza en la app de demostración [Sunflower](https://github.com/googlesamples/android-sunflower).

Videos

<https://www.youtube.com/watch?v=XamMbnzI5vE&feature=emb_logo> -- > Building interfaces with ConstraintLayout

<https://www.youtube.com/watch?v=OHcfs6rStRo&feature=emb_logo> -- > Keyframes animations with ConstraintLayout