



Aplicaciones Móviles

DOMICIANO RINCÓN

INGENIERÍA TELEMÁTICA
DISEÑO DE MEDIOS INTERACTIVOS
INGENIERÍA DE SISTEMAS



Estructura de una App

ANDROID

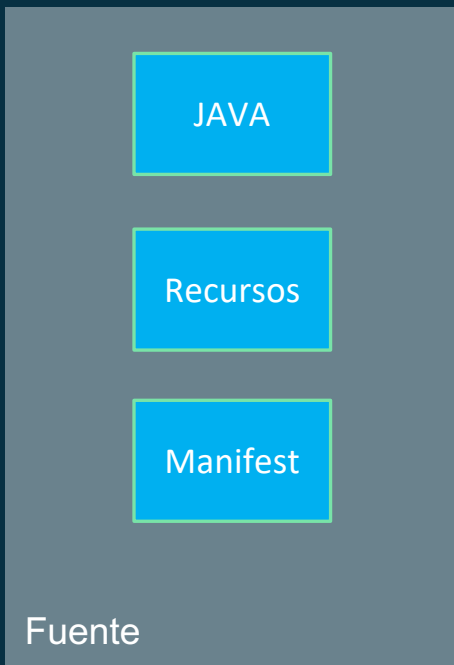
Aplicaciones Android



APK

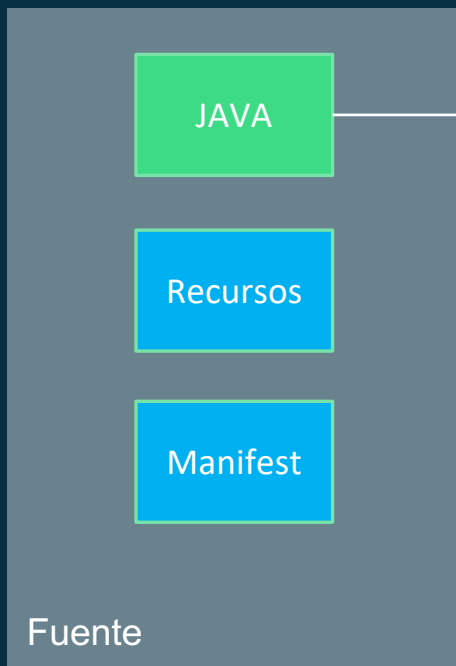
El compilador empaqueta los componentes de la aplicación en un archivo de extensión .APK.

Aplicaciones Android



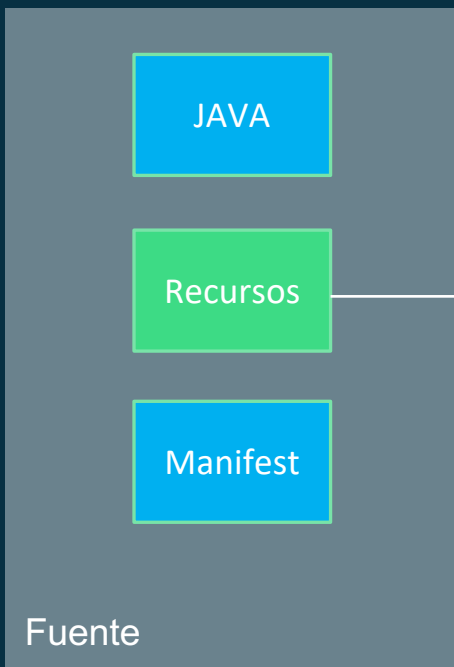
El compilador empaqueta los componentes de la aplicación en un archivo de extensión .APK.

Aplicaciones Android



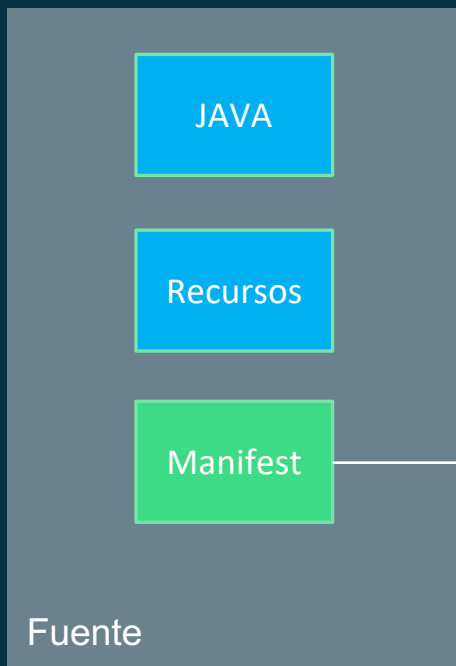
Archivos donde irán los algoritmos necesarios para la ejecución programada de nuestra aplicación. Son kotlin, pero la carpeta se llama Java

Aplicaciones Android



Donde se encuentran las vistas de la aplicación, imágenes, iconos, colores, texto y constantes que se podrán usar en la parte JAVA

Aplicaciones Android



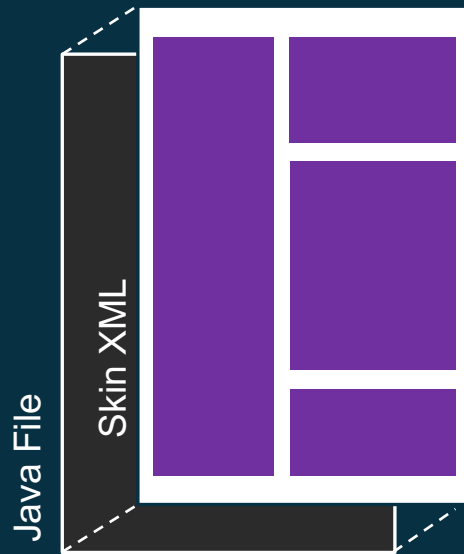
Un archivo XML que presenta la configuración de la aplicación, donde se define el nombre, el icono y la clase de JAVA a ejecutar al accionar la aplicación, los permisos concedidos a la app, los servicios que ejecuta, canales de comunicación con otras aplicaciones entre otros.

Activity

ANDROID

Activity

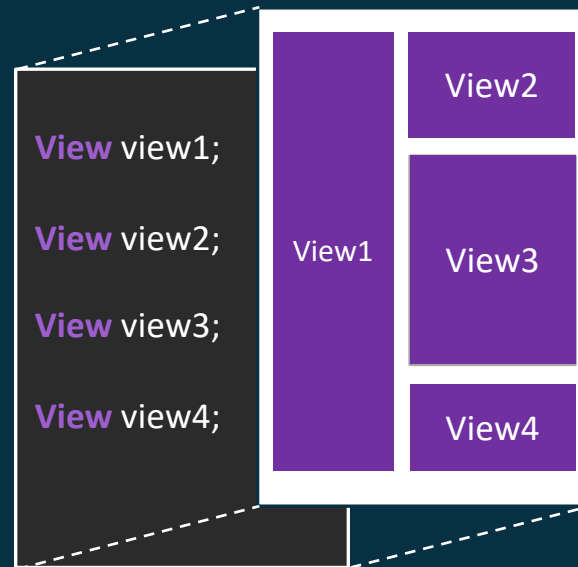
«Una actividad es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción, como marcar un número telefónico, tomar una foto, enviar un correo electrónico o ver un mapa. A cada actividad se le asigna una ventana en la que se puede dibujar su interfaz de usuario»



Activity

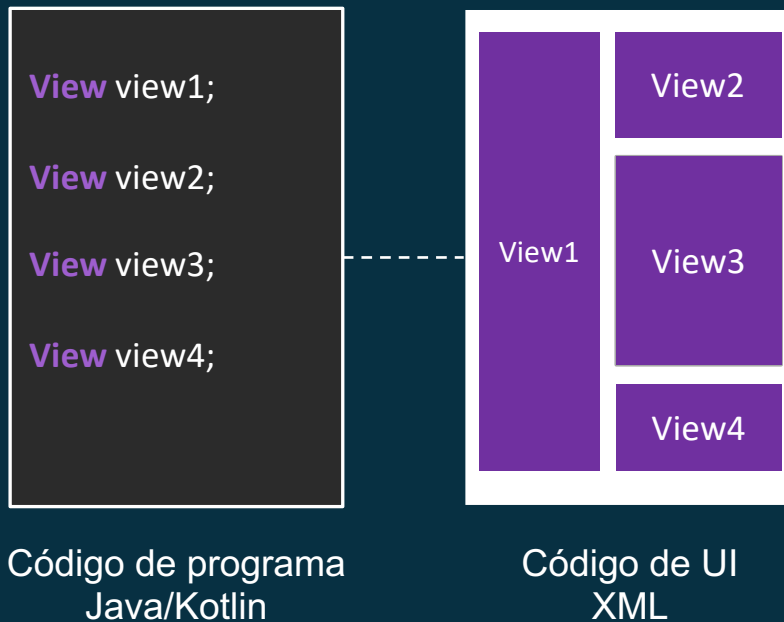
Activity

«Una actividad es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción, como marcar un número telefónico, tomar una foto, enviar un correo electrónico o ver un mapa. A cada actividad se le asigna una ventana en la que se puede dibujar su interfaz de usuario»

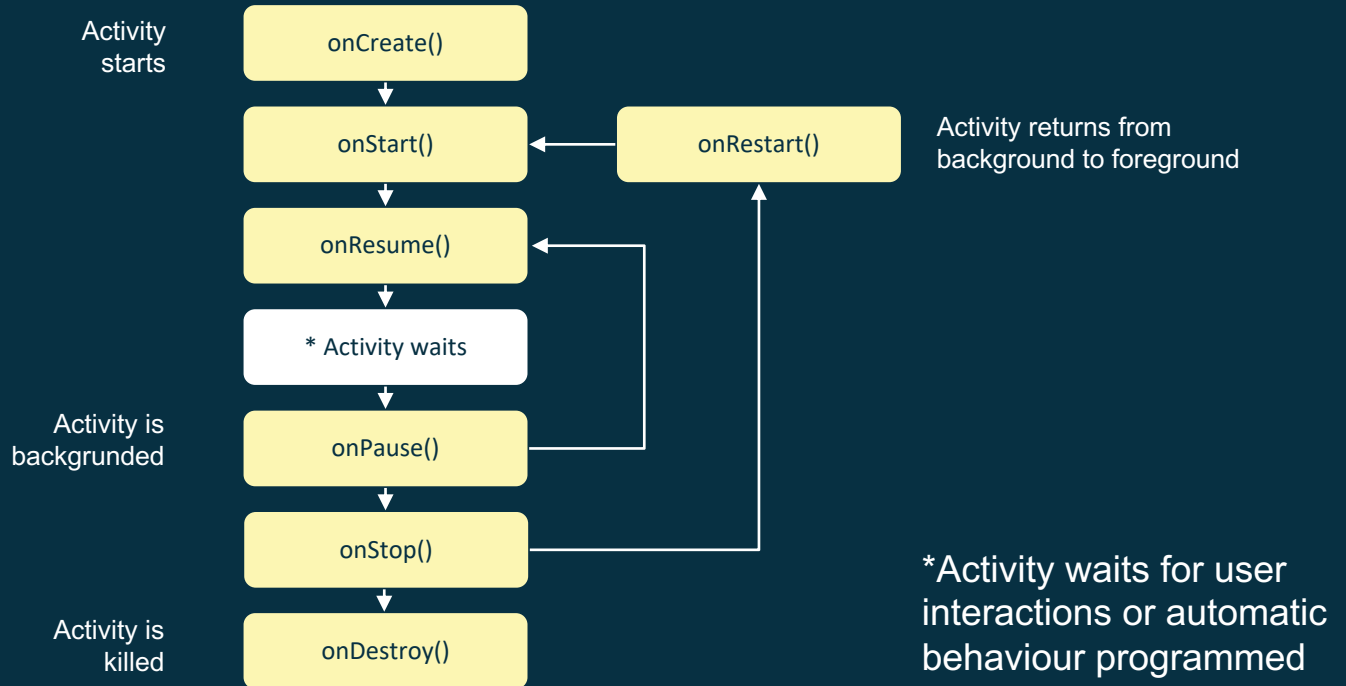


Activity

Activity



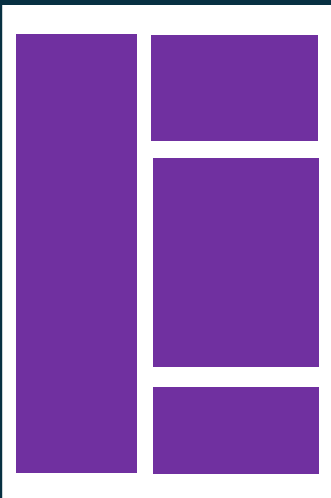
Activity: ciclo de vida



Diseño de views

XML UI

Views



Código de UI
XML

La organización de los elementos en Android tiene unas reglas básicas:

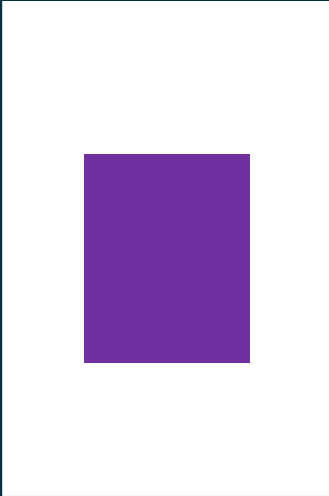
Se define al menos una margen horizontal

Se define al menos una margen vertical

Se define el ancho del elemento

Se define el alto del elemento

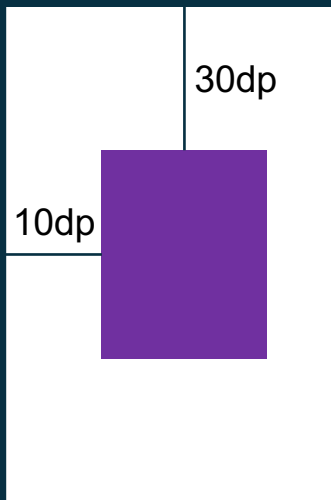
Views



Diseño de UI

Por ejemplo para ubicar el elemento morado de la izquierda debe definir margen vertical y horizontal

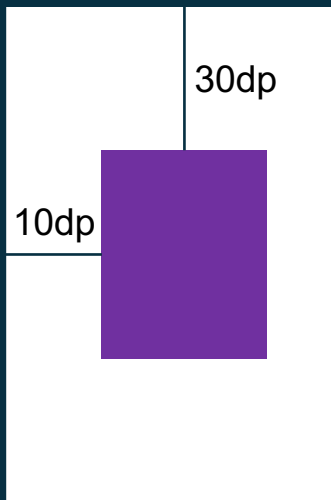
Views



Diseño de UI

Por ejemplo para ubicar el elemento morado de la izquierda debe definir margen vertical y horizontal

Views

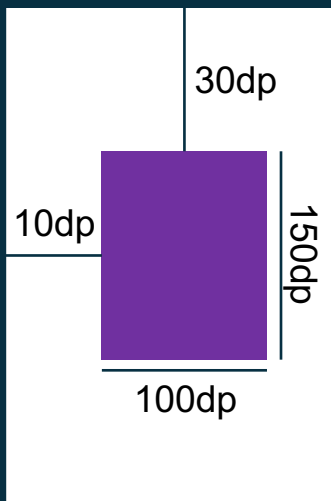


Diseño de UI

Por ejemplo para ubicar el elemento morado de la izquierda debe definir margen vertical y horizontal

Luego se define el ancho y el alto

Views



Diseño de UI

Por ejemplo para ubicar el elemento morado de la izquierda debe definir margen vertical y horizontal

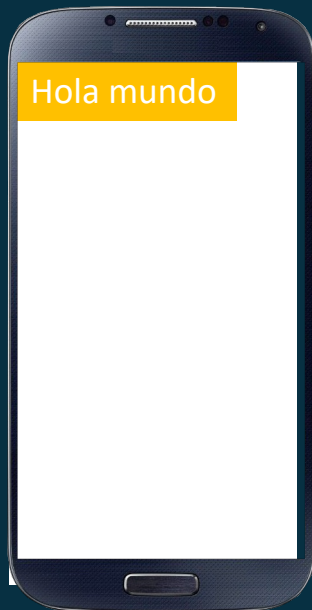
Luego se define el ancho y el alto

Dimensiones

XML UI

Dimensiones

El contenido del view es el texto. Para los Layouts, el contenido son los views que contiene



Width

El ancho del view
`w="wrap_content"`

Height

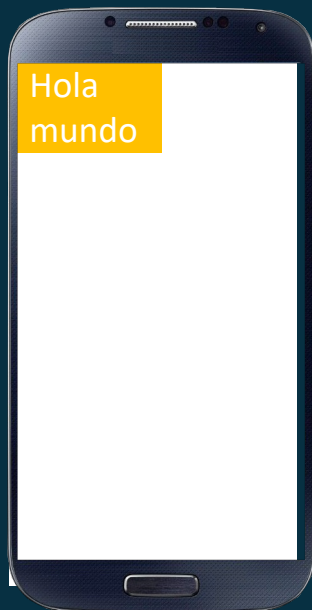
El alto del view
`h="wrap_content"`

wrap_content

El view se ajusta al contenido

Dimensiones – Wrap Content

Usar esta dimensión en lugar de un número implica que el elemento va a ser del ancho y alto necesarios para presentar su contenido



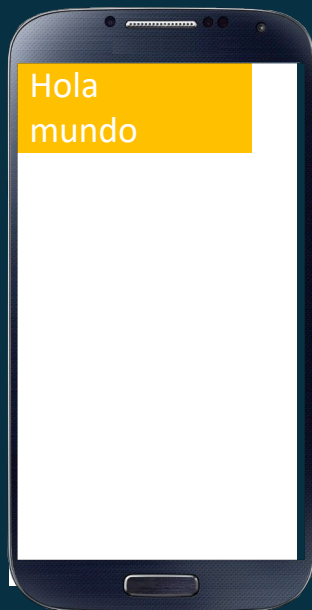
Width en wrap_content

Height en wrap_content

wrap_content
El view se ajusta al contenido

Dimensiones – Wrap Content

Usar esta dimensión en lugar de un número implica que el elemento va a ser del ancho y alto necesarios para presentar su contenido



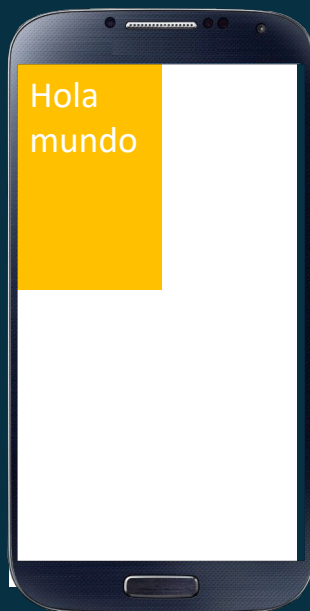
Width en 300dp

Height en wrap_content

wrap_content
El view se ajusta
al contenido

Dimensiones – Wrap Content

Usar esta dimensión en lugar de un número implica que el elemento va a ser del ancho y alto necesarios para presentar su contenido



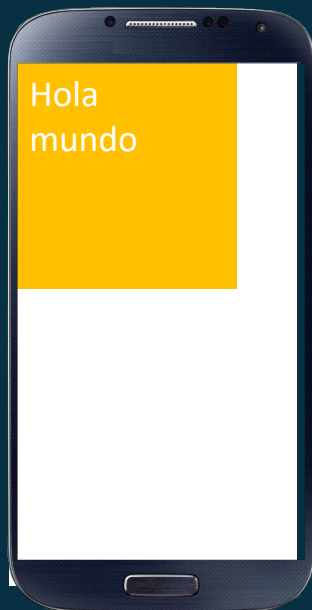
Width en wrap_content

Height en 400dp

wrap_content
El view se ajusta
al contenido

Dimensiones – Wrap Content

Usar esta dimensión en lugar de un número implica que el elemento va a ser del ancho y alto necesarios para presentar su contenido



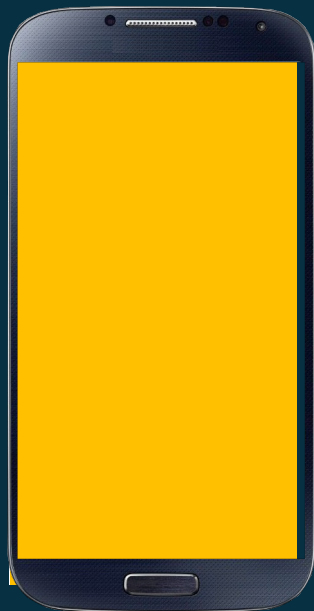
Width en 300dp

Height en 400dp

wrap_content
El view se ajusta al contenido

Dimensiones – Match Parent

match_parent implica que el elemento va a usar el espacio disponible según el eje para expandirse



Width en match_parent

Height en match_parent

match_parent
El view se
expande en el eje

Dimensiones – Match Parent

match_parent implica que el elemento va a usar el espacio disponible según el eje para expandirse



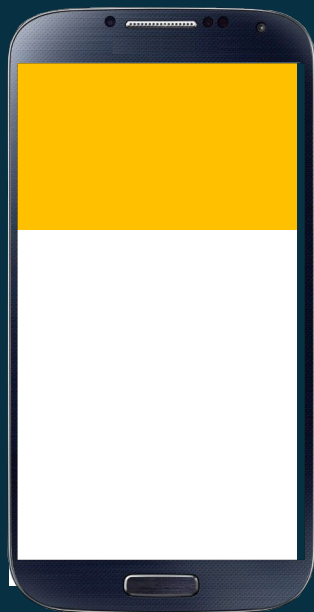
Width en 200dp

Height en match_parent

match_parent
El view se
expande en el eje

Dimensiones – Match Parent

match_parent implica que el elemento va a usar el espacio disponible según el eje para expandirse



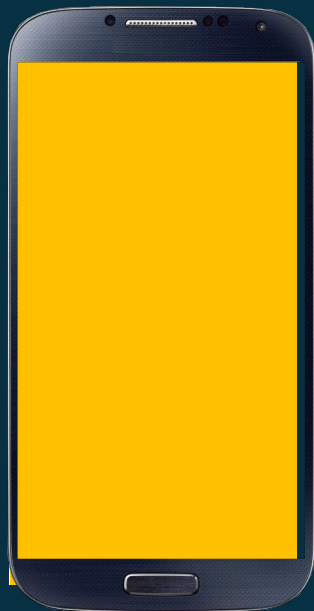
Width en match_parent

Height en 200dp

match_parent
El view se
expande en el eje

Dimensiones – Match Parent

match_parent implica que el elemento va a usar el espacio disponible según el eje para expandirse



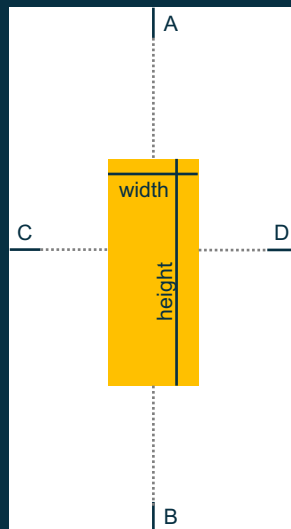
Width en match_parent

Height en match_parent

Es un view que se usa sobre todo para el padre de todos los elementos o root layout

Dimensiones – Match Constraint

match_constraint implica que el elemento va a usar el espacio disponible respetando las márgenes impuestas



Width en 100dp

Height en 300dp

Margen A mide 20dp

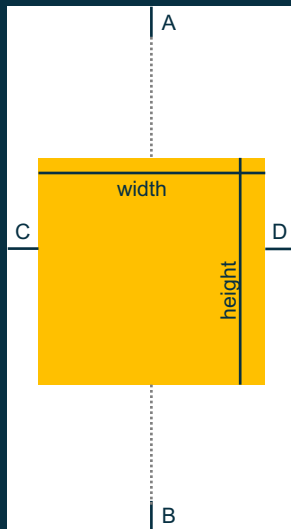
Margen B mide 20dp

Margen C mide 20dp

Margen D mide 20dp

Dimensiones – Match Constraint

match_constraint implica que el elemento va a usar el espacio disponible respetando las márgenes impuestas



Width en match_constraint

Height en 300dp

Margen A mide 20dp

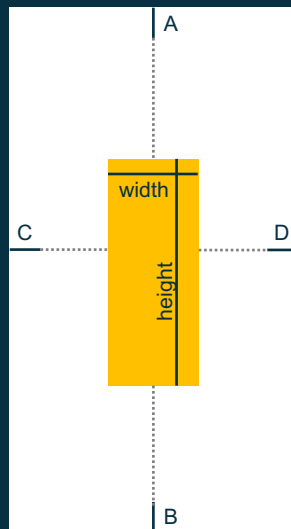
Margen B mide 20dp

Margen C mide 20dp

Margen D mide 20dp

Dimensiones – Match Constraint

match_constraint implica
que el elemento va a usar
el espacio disponible
respetando las márgenes
impuestas



Width en 100dp

Height en 300dp

Margen A mide 20dp

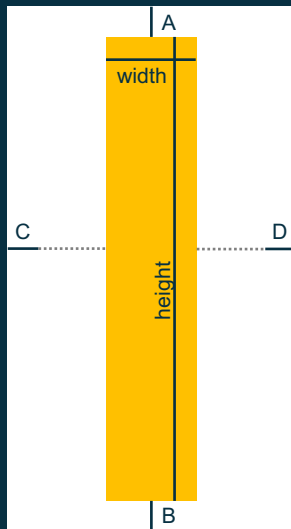
Margen B mide 20dp

Margen C mide 20dp

Margen D mide 20dp

Dimensiones – Match Constraint

match_constraint implica que el elemento va a usar el espacio disponible respetando las márgenes impuestas



Width en 100dp

Height en match_constraints

Margen A mide 20dp

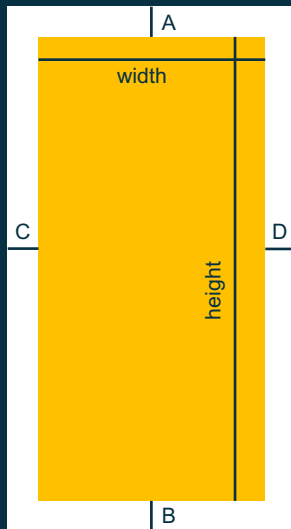
Margen B mide 20dp

Margen C mide 20dp

Margen D mide 20dp

Dimensiones – Match Constraint

match_constraint implica que el elemento va a usar el espacio disponible respetando las márgenes impuestas



Width en match_constraints

Height en match_constraints

Margen A mide 20dp

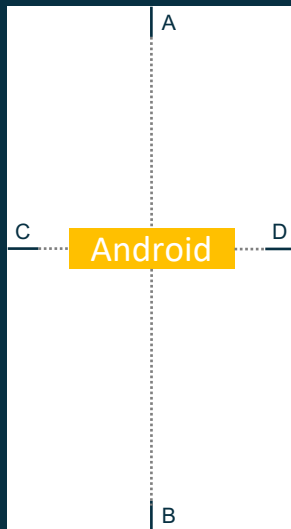
Margen B mide 20dp

Margen C mide 20dp

Margen D mide 20dp

Dimensiones

Si por ejemplo, el elemento tiene un contenido, también puede ajustar las dimensiones verticales y horizontales a `wrap_content`



Width en wrap_content

Height en wrap_content

Margen A mide 20dp

Margen B mide 20dp

Margen C mide 20dp

Margen D mide 20dp