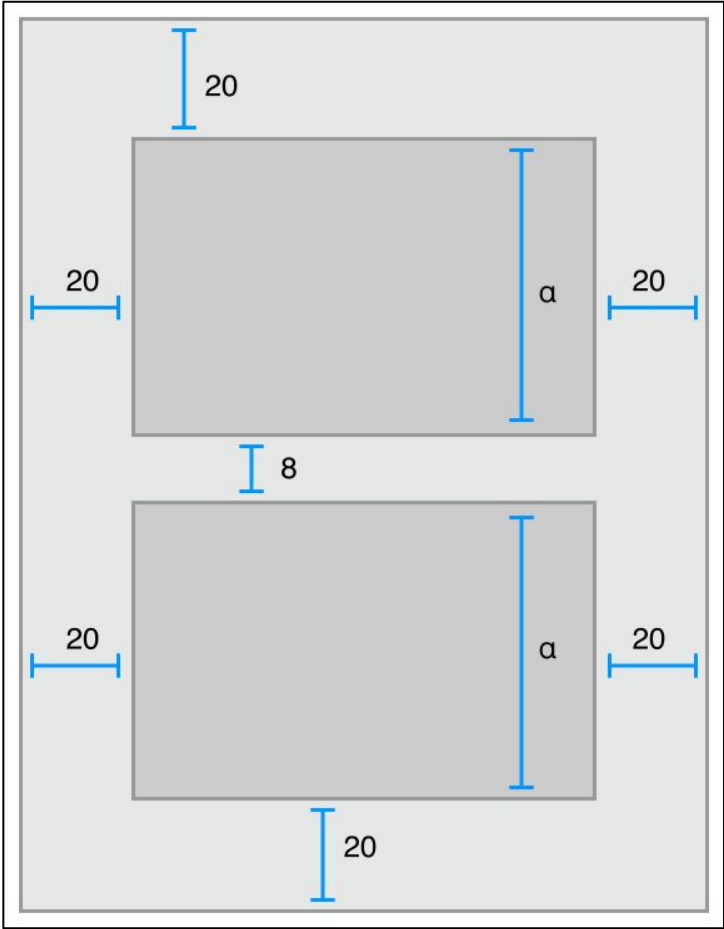
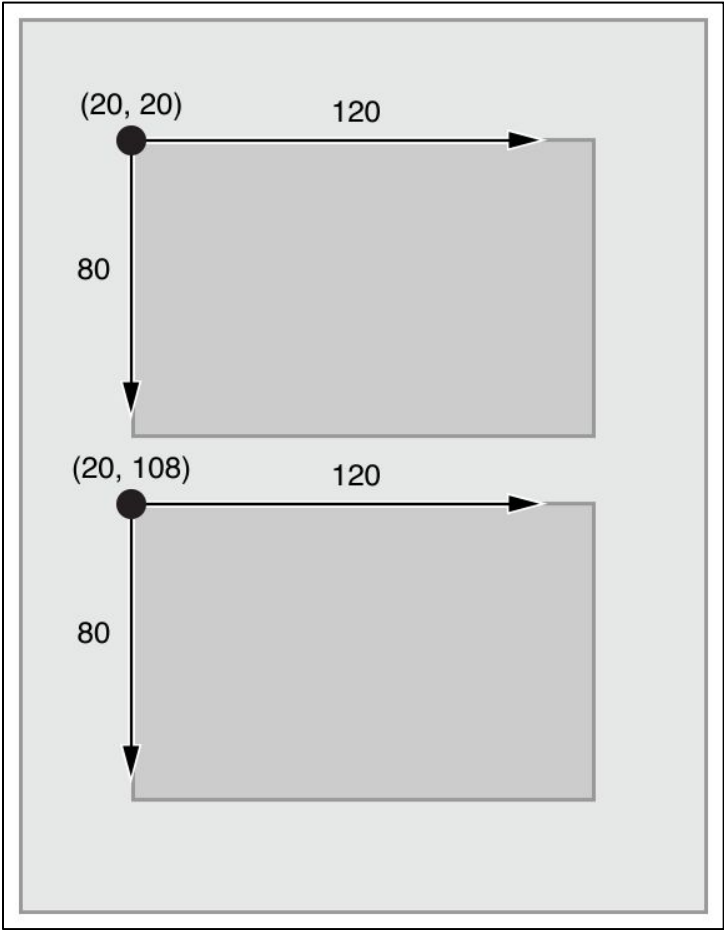


AUTOLAYOUT





Auto Layout define la interfaz de usuario mediante una serie de **restricciones**.

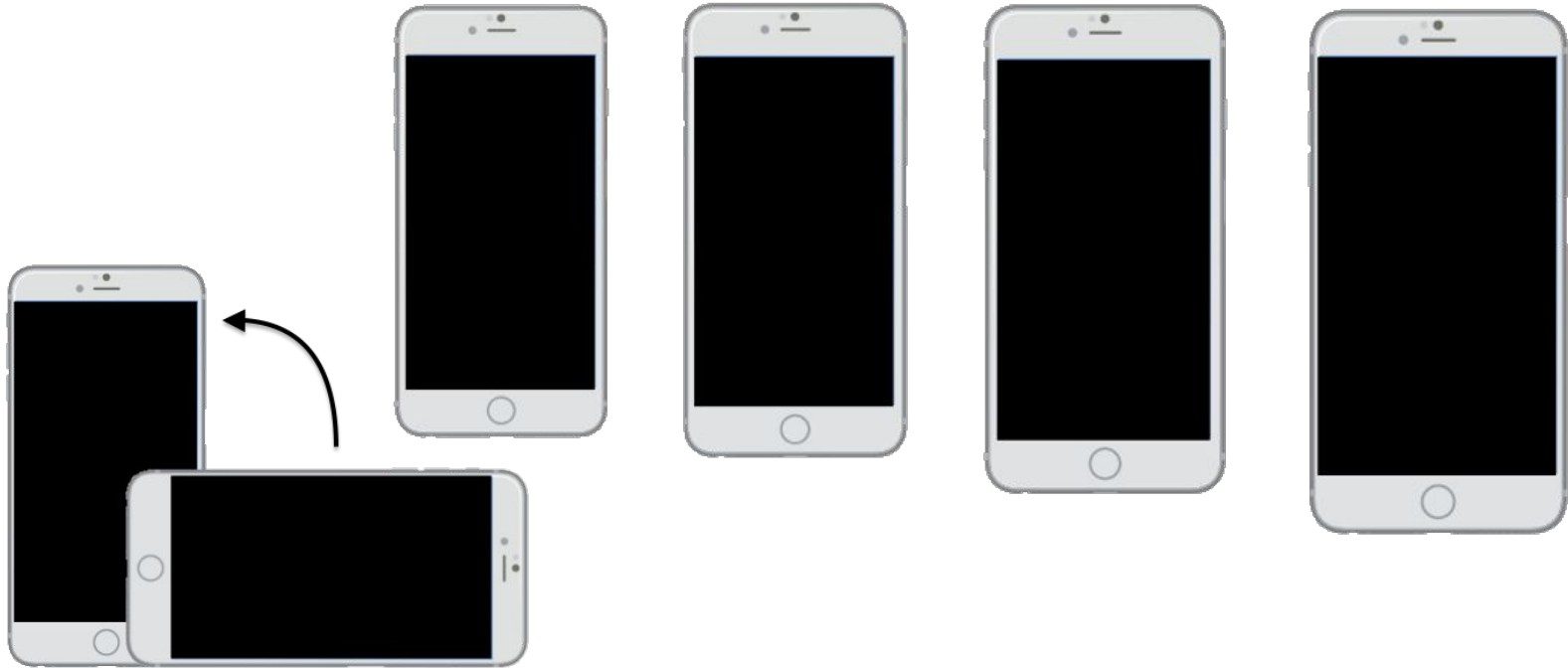
Las restricciones típicamente representan una **relación** entre dos vistas.

Luego, en base a esas relaciones, se calcula el **tamaño** y **posición** de cada vista.

Este mecanismo genera diseños (layouts) que responden dinámicamente a **cambios** tanto internos como externos.

[illegible]

Cambios **externos** pueden deberse a que el dispositivo fue rotado o a que se soportan distintos tamaños de pantalla.



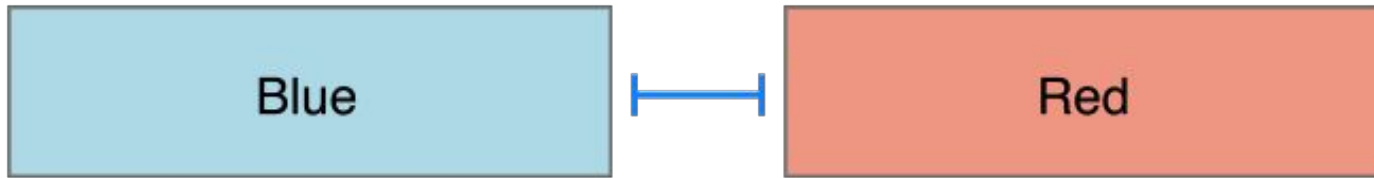
Auto Layout define la interfaz de usuario mediante una serie de **restricciones (CONSTRAINTS)**.

Cada restricción representa una **ecuación lineal**, una representación matemática de una regla.

Todas las restricciones deben eventualmente combinarse en un sistema de ecuaciones que tiene una, y sólo una, **solución posible**.

AUTO LAYOUT

“El **borde izquierdo** de la **vista roja** debe ubicarse **8 puntos** a la derecha del **borde derecho** de la **vista azul**.”



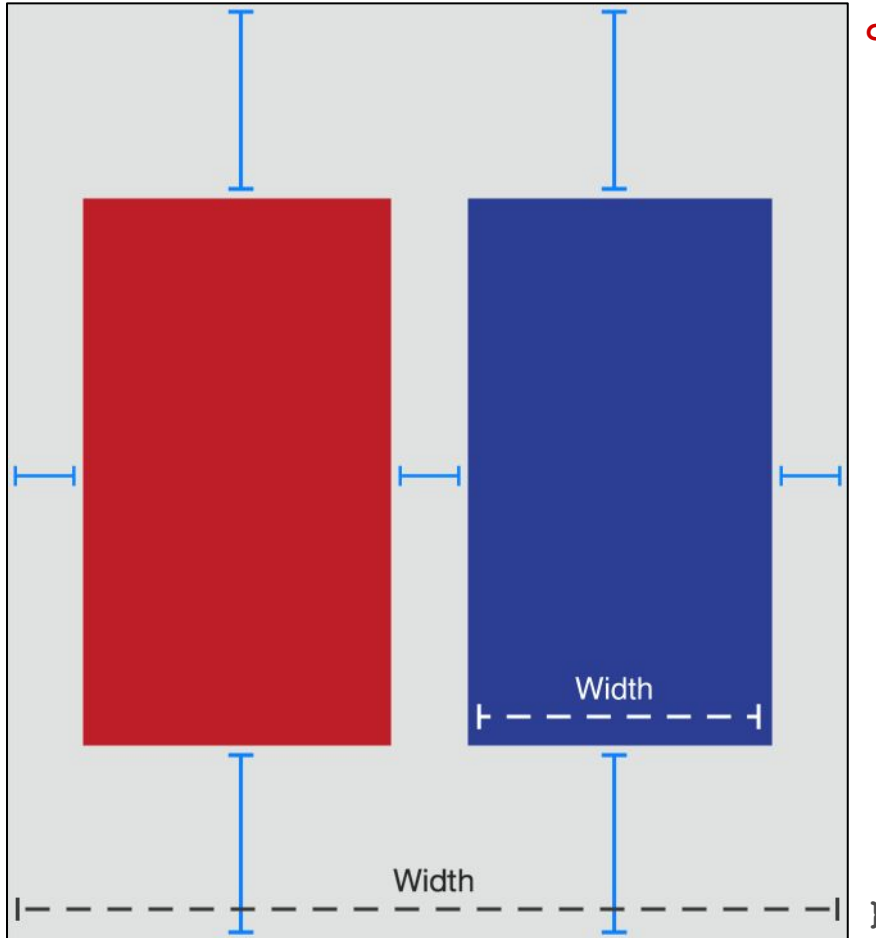
$$\text{RedView.Leading} = 1.0 \times \text{BlueView.trailing} + 8.0$$

Diagram illustrating the constraint equation with labels for its components:

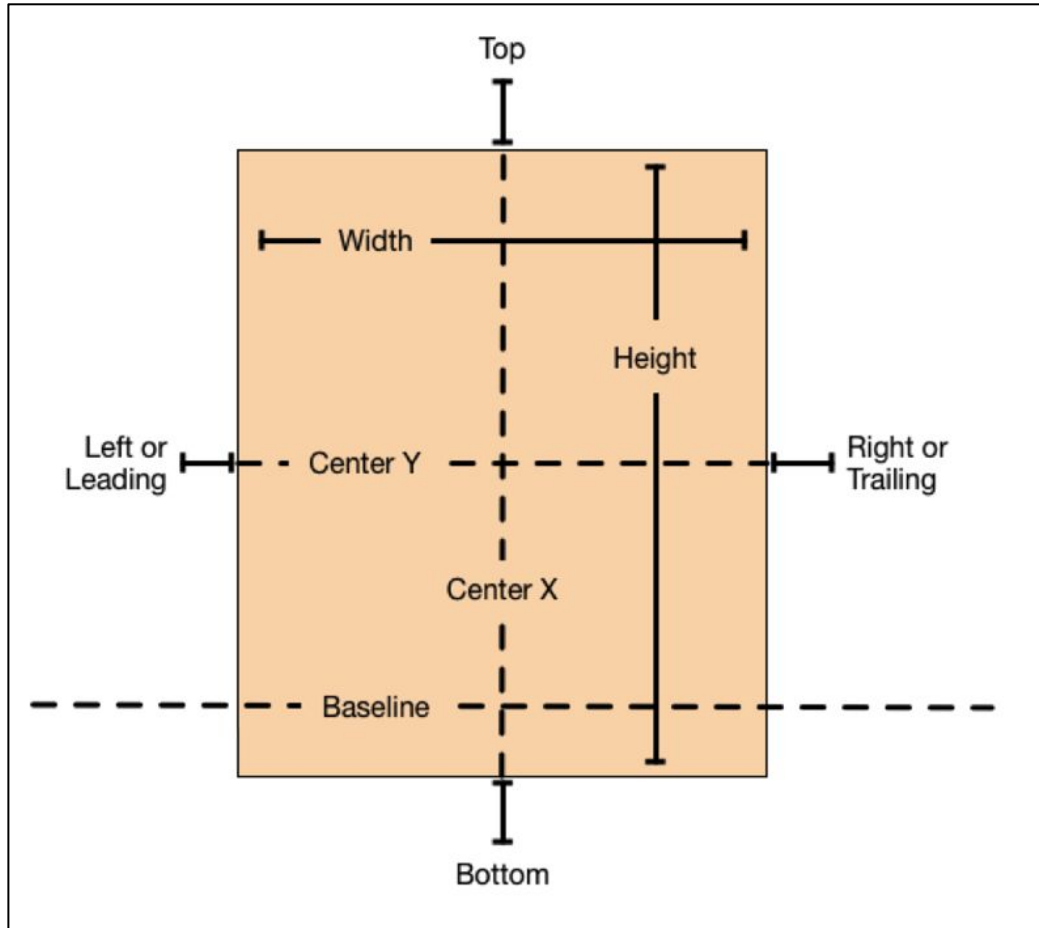
- RedView.Leading**: Labeled as **Item 1**.
- =**: Labeled as **Relationship**.
- 1.0**: Labeled as **Multiplier**.
- BlueView.trailing**: Labeled as **Item 2**.
- +**: Labeled as **Relationship**.
- 8.0**: Labeled as **Constant**.

Additional labels for the terms in the equation:

- BlueView.trailing** is also labeled as **Attribute 1**.
- 8.0** is also labeled as **Attribute 2**.



```
class NSLayoutConstraint {  
    var firstItem: AnyObject  
    var firstAttribute: NSLayoutConstraintAttribute  
    var secondItem: AnyObject? // optional  
    var secondAttribute: NSLayoutConstraintAttribute  
  
    var multiplier: CGFloat  
    var constant: CGFloat  
  
    var relation: NSLayoutConstraintRelation  
    1. lessThanOrEqualTo  
    2. greaterThanOrEqualTo  
    3. equal  
  
    var priority: UILayoutPriority // number  
    // constraints with higher priority are  
    // satisfied before ones with lower prio  
}
```

Atributos:

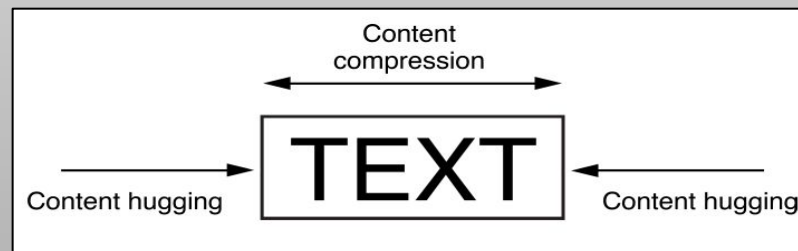
- left
- right
- top
- bottom
- leading (left/right)
- trailing (left/right)
- width
- height
- centerX
- centerY
- baseline

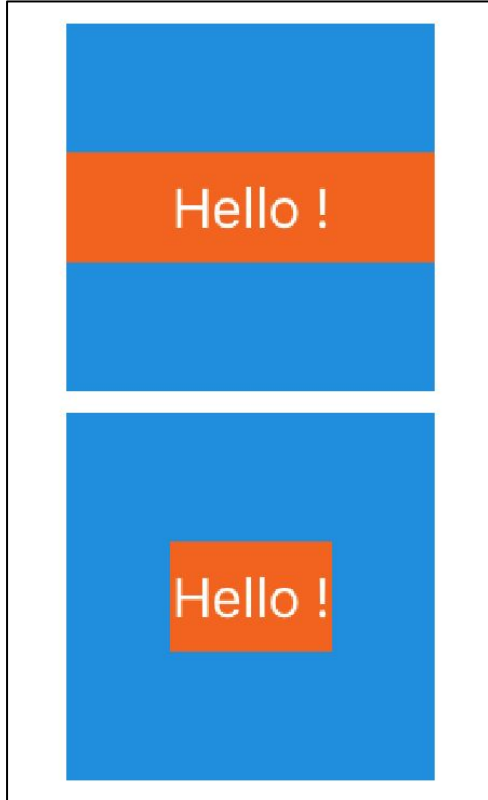
INTRINSIC CONTENT SIZE

Es el **tamaño natural** de una vista
dado su contenido.

Si una vista tiene intrinsic content size, y no
tiene restricciones de ancho/alto, Auto
Layout la va a ajustar a su intrinsic content
size.

Por ejemplo, el intrinsic content size de un botón es el
tamaño de su título más un pequeño margen.





Content **Hugging** Priority

El botón naranja es **más chico** que la vista azul.
Los bordes del botón naranja están alineados a los bordes de la vista azul con **prioridad = 500**.

SUPERIOR

Horizontal Hugging Priority es **750 (mayor que 500)**.
(el botón ocupa **tanto espacio como sea posible**)

INFERIOR

Horizontal Hugging Priority es **250 (menor que 500)**.
(el botón ocupa **el menor espacio posible**)



Content **Compression Resistance** Priority

El botón naranja es **más grande** que la vista azul.
Los bordes del botón naranja están alineados a los bordes de la vista azul con **prioridad = 500**.

SUPERIOR

Horizontal Compression Resistance Priority es **750 (mayor que 500)**.
(el botón *rehúsa* ser comprimido, **resiste la compresión**)

INFERIOR

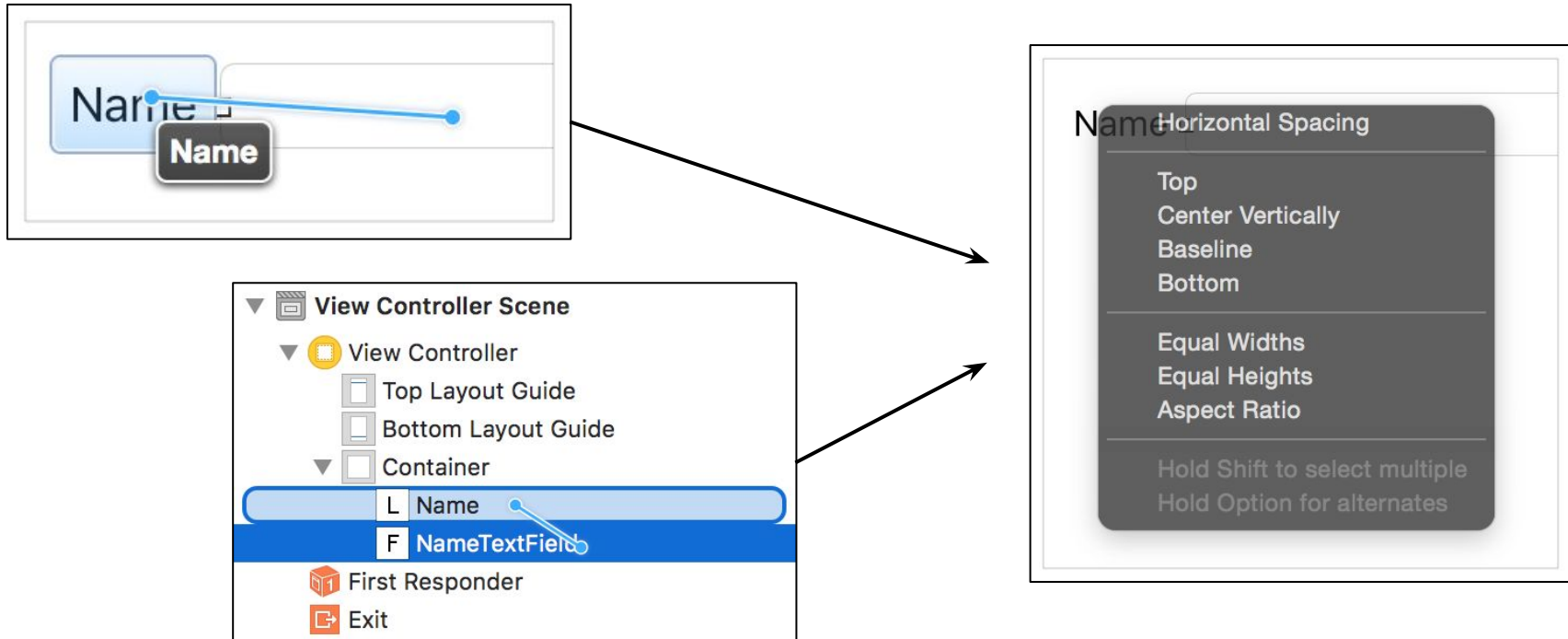
Horizontal Compression Resistance Priority es **250 (menor que 500)**.
(el botón *permite* ser comprimido, **NO resiste la compresión**)

AUTO LAYOUT EN XCODE

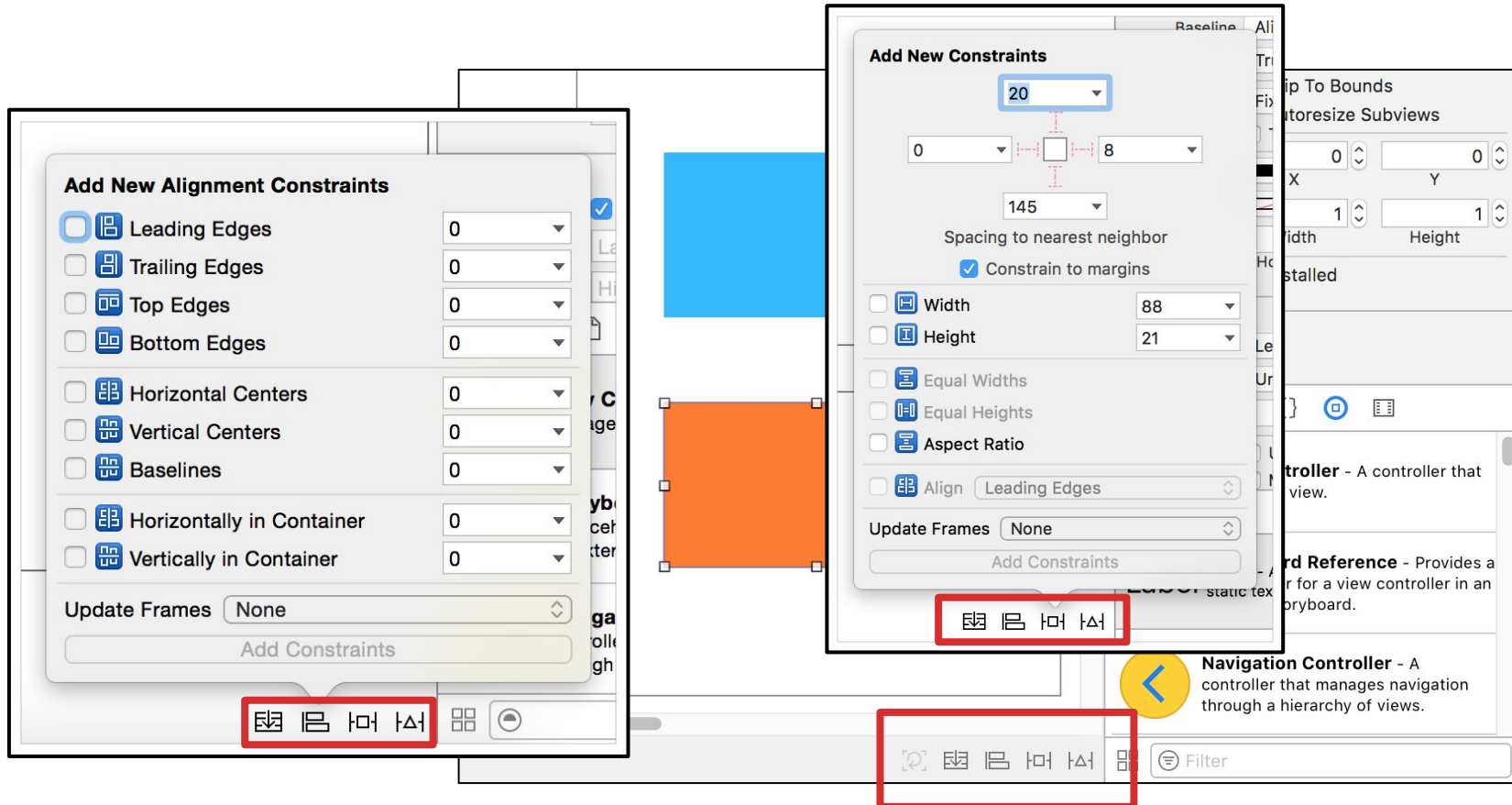
Hay dos alternativas para **configurar** constraints de Auto Layout en Interface Builder.

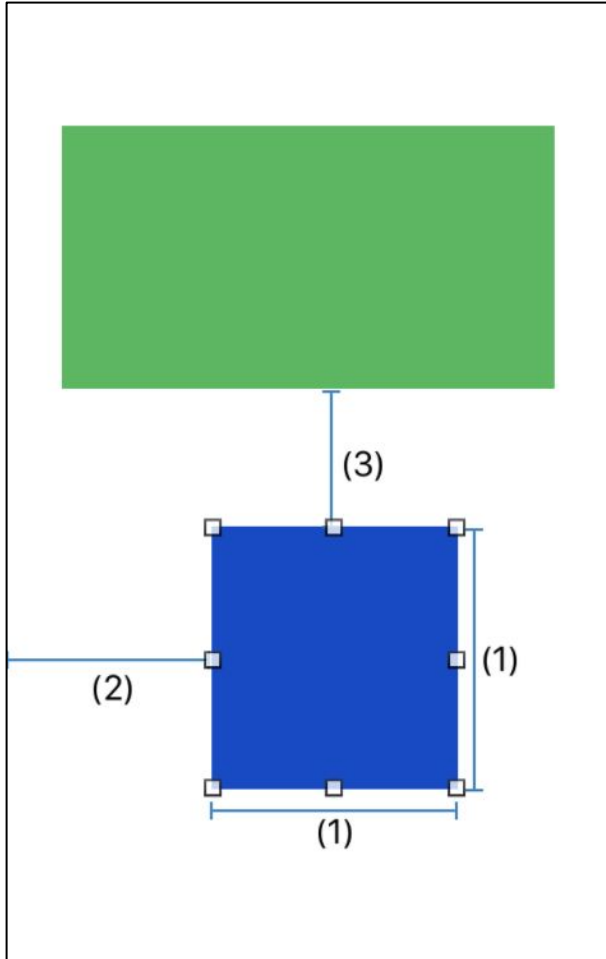
Y tres **posibles constraints** para configurar.

1. Configurar una constraint haciendo **ctrl-drag** (o **right-click-drag**) desde una vista a otra (o a sí misma).
Luego elegir la constraint a configurar.



2. Configurar una constraint usando la **Pin & Align Tool**.



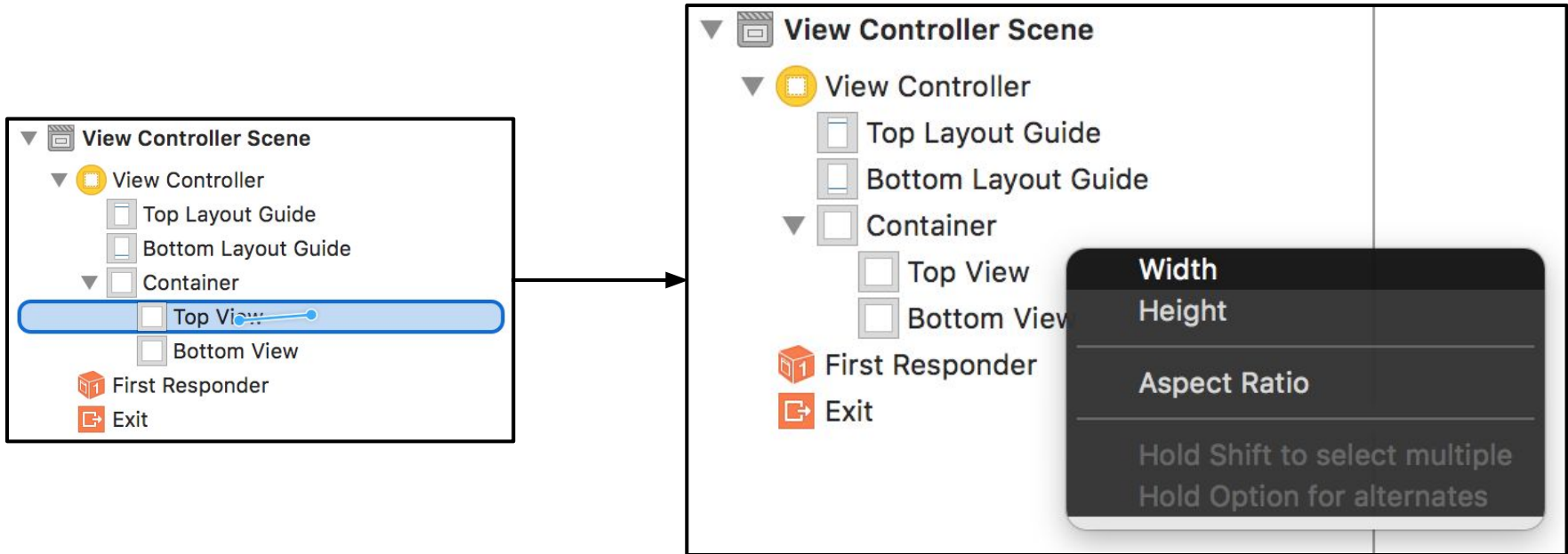


Cuando se **configuran constraints** de Auto Layout en el Interface Builder, hay tres posibilidades o alternativas:

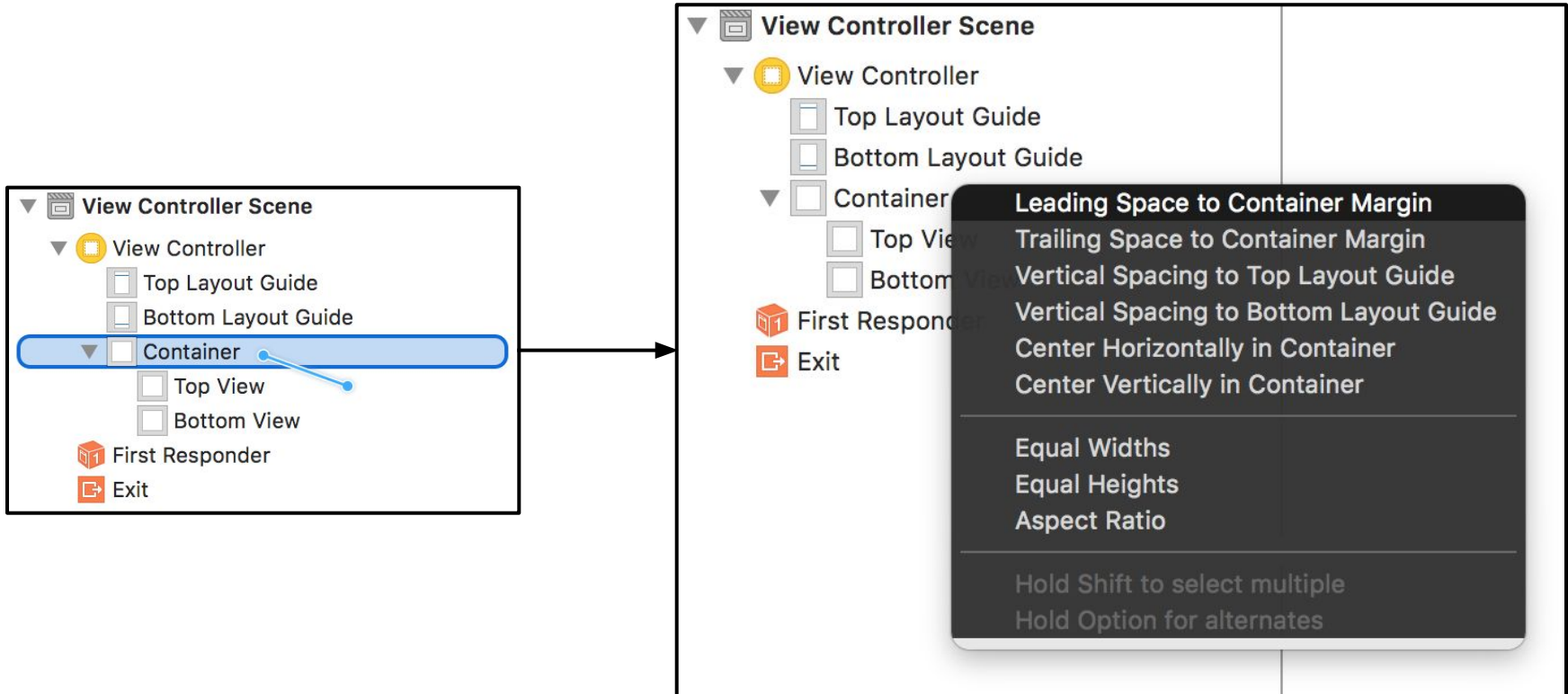
1. Constraint de una vista **consigo misma**.
2. Constraint de una vista **con su contenedor**.
3. Constraint de una vista **con un hermano**.

(una constraint entre dos vistas contenidas por el mismo contenedor)

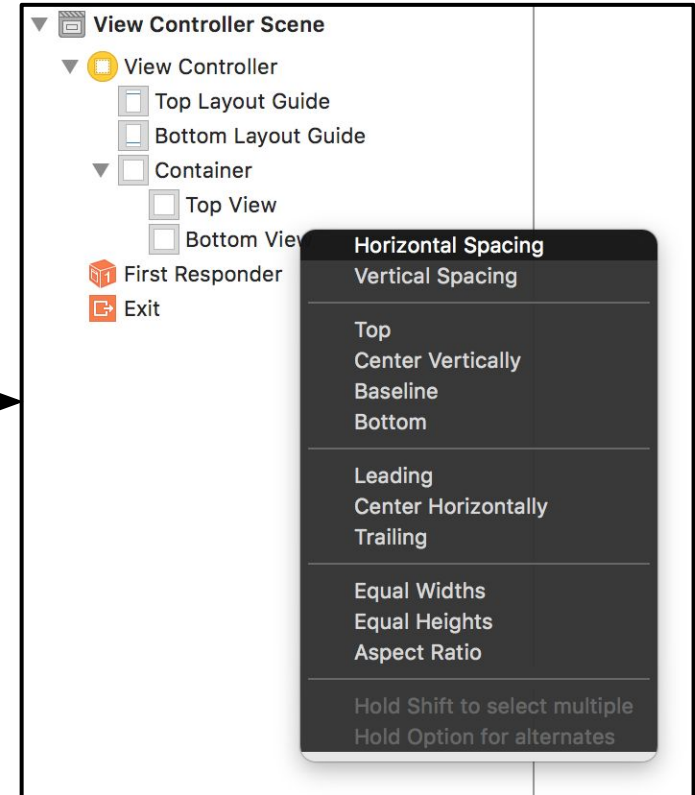
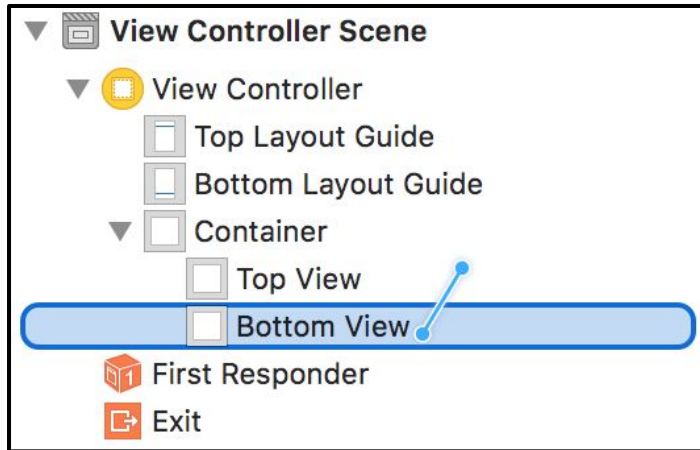
1. Constraint de una Vista **consigo misma**.



2. Constraint de una Vista **con su Contenedor**.



3. Constraint de una Vista **con un Hermano**.



Los errores en tiempo de ejecución son, por lo general, muy crípticos y es difícil entender donde está el problema.

MANEJO DE ERRORES

Para ello, se recomienda el sitio <https://www.wtfautolayout.com/> donde, dado un log de error, se pueden ver las constraints que están generando conflicto.