

¿Por qué TypeScript?



NO ES OBLIGATORIO

No necesitas TypeScript para React Native



JS

¿Por qué TypeScript?

JS

```
const persona = {
  primerNombre: 'Fernando',
  primerApellido: 'Herrera'
  fechaNacimiento: 12345627381293
}

console.log( persona.primerApellido );
```

TS

```
const persona = {
  primerNombre: 'Fernando',
  primerApellido: 'Herrera'
  fechaNacimiento: 12345627381293
}

console.log( persona.primerApellido );
```

{dev/talles}

```
const calculaSalario = ( empleado ) => {
```

```
> undefined > 5
< false
```

```
  const nuevoSalario = empleado.antiguedad > 5
    ? empleado.salario + 250
    : empleado.salario + 150;
```

```
  return {
    empleado,
    nuevoSalario
  }
}
```

```
const empleado = {
  id: 1,
  nombre: 'Fernando',
  sueldo: '1500'
}
```



```
calculaSalario( empleado );
```

```
const calculaSalario = ( empleado: Empleado ) => {
```

```
  const nuevoSalario = empleado.antiguedad > 5
    ? empleado.salario + 250
    : empleado.salario + 150;
```

```
  return {
    empleado,
    nuevoSalario
  }
}
```

```
const empleado = {
  id: 1,
  nombre: 'Fernando',
  sueldo: '1500'
}
```



```
calculaSalario( empleado );
```

```
const calculaSalario = ( empleado: Empleado ) => {

  const nuevoSalario = empleado.antiguedad > 5
    ? empleado.salario + 250
    : empleado.salario + 150;

  return {
    empleado,
    nuevoSalario
  }
}

const empleado = {
  id: 1,
  nombre: 'Fernando',
  sueldo: '1500'
}

calculaSalario( empleado );
```



```
const empleado: {
  id: number;
  nombre: string;
  sueldo: string;
}

Argument of type '{ id: number; nombre: string; sueldo: string; }' is not
assignable to parameter of type 'Empleado'.
Type '{ id: number; nombre: string; sueldo: string; }' is missing the following
properties from type 'Empleado': salario, antiguedad ts(2345)
Peek Problem (^F8) No quick fixes available

(empleado );
```

Características Principales

- Tipado estricto.
- Mutación controlada (en caso de ser necesario).
- Documentación automática de funciones, argumentos y métodos.
- Control null, undefined, NaN, etc.
- Ayuda enormemente cuando se trabaja en equipo.

Problemas comunes en TypeScript

- No sé qué tipo de dato poner.
- No sé qué argumento recibo.
- El objeto que se recibe tiene extiende de otro, ¿Qué hago?

Mito: **TypeScript le quita la flexibilidad que JavaScript me ofrece**

Antes de comenzar el curso:

Por favor, les recomiendo que sigan el curso con TypeScript, verán el beneficio directo de un tipado estricto.

Déjenme el estrés de los tipos a mi :)

**Absoluta, relativa, top,
left, right, bottom**

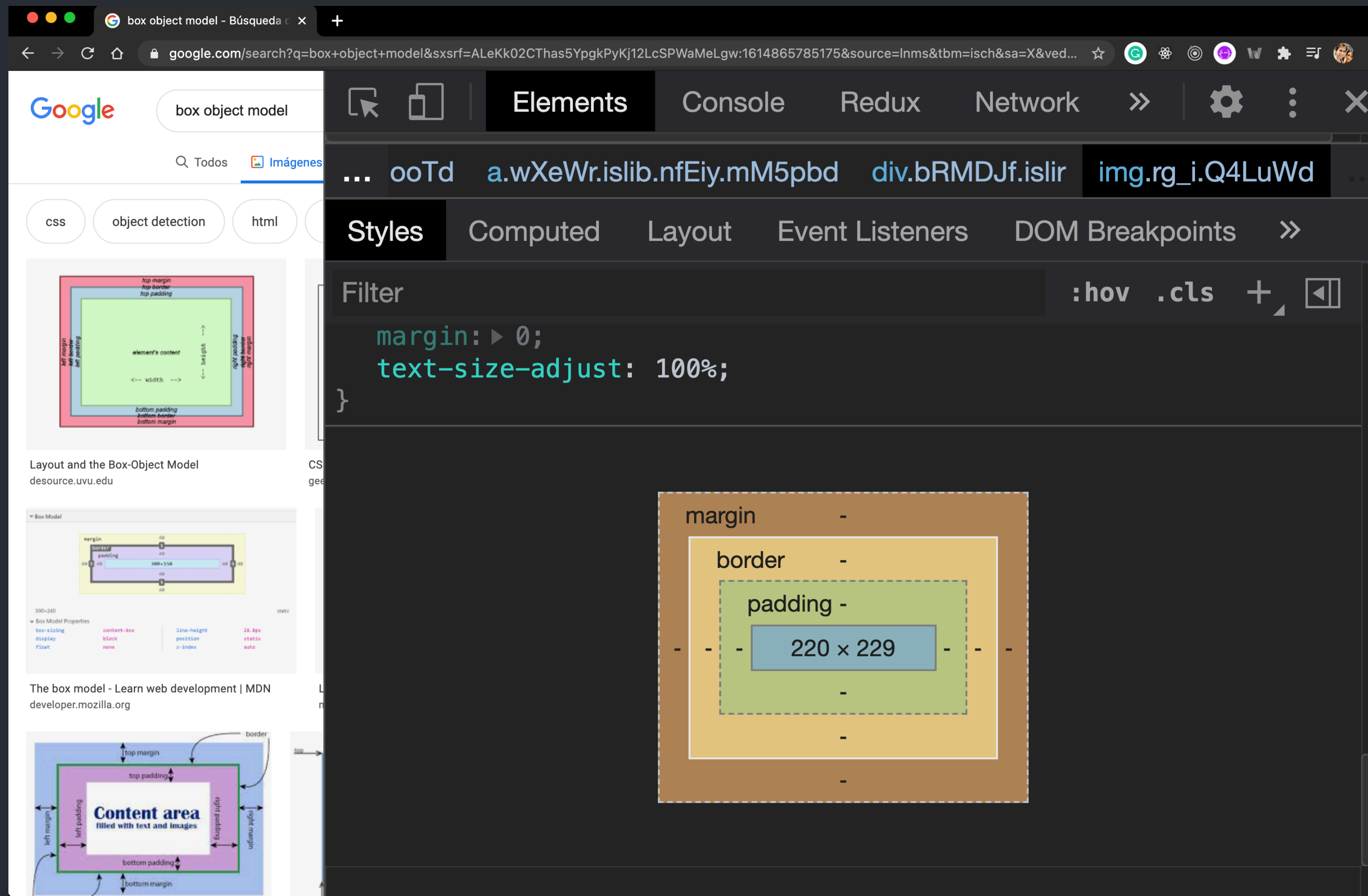
Box Object Model

**Alto, ancho, margen,
padding, borde.**

Position

Flex Box

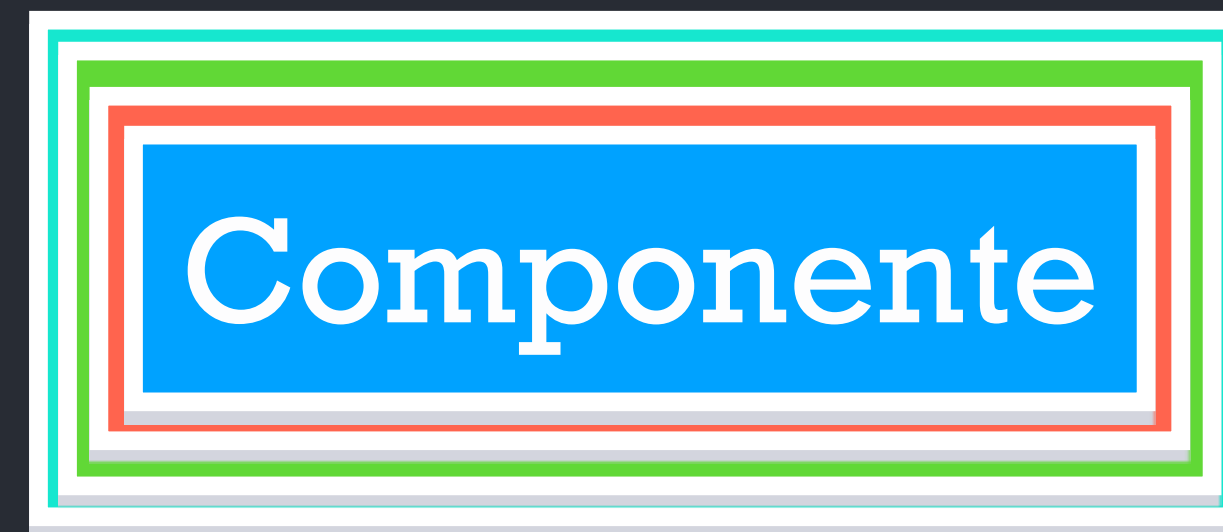
**Dirección, ubicación,
alineamiento, estirar,
encoger, proporciones.**



Margin: 0

Padding: 0

Border: 0



Realidad

Propiedad	Descripción
margin	Aplica margen a los 4 lados
marginLeft	Margen a la izquierda
marginRight	Margen a la derecha
marginBottom	Margen abajo
marginTop	Margen arriba
marginVertical	Aplica mismo margen arriba y abajo
marginHorizontal	Mismo margen a la izquierda y derecha

Margin

Propiedad	Descripción
margin	Aplica margen a los 4 lados
marginLeft	Margen a la izquierda
marginRight	Margen a la derecha
marginBottom	Margen abajo
marginTop	Margen arriba
marginVertical	Aplica mismo margen arriba y abajo
marginHorizontal	Mismo margen a la izquierda y derecha

Padding

Propiedad	Descripción
padding	Aplica padding a los 4 lados
paddingLeft	Padding a la izquierda
paddingRight	Padding a la derecha
paddingBottom	Padding abajo
paddingTop	Padding arriba
paddingVertical	Aplica mismo padding arriba y abajo
paddingHorizontal	Mismo padding a la izquierda y derecha

Border

Propiedad	Descripción
borderWidth	Aplica borde a los 4 lados
borderLeftWith	Borde a la izquierda
borderRightWith	Borde a la derecha
borderBottomWith	Borde abajo
borderTopWith	Borde arriba

borderColor: ‘black’

**Absoluta, relativa, top,
left, right, bottom**

Box Object Model

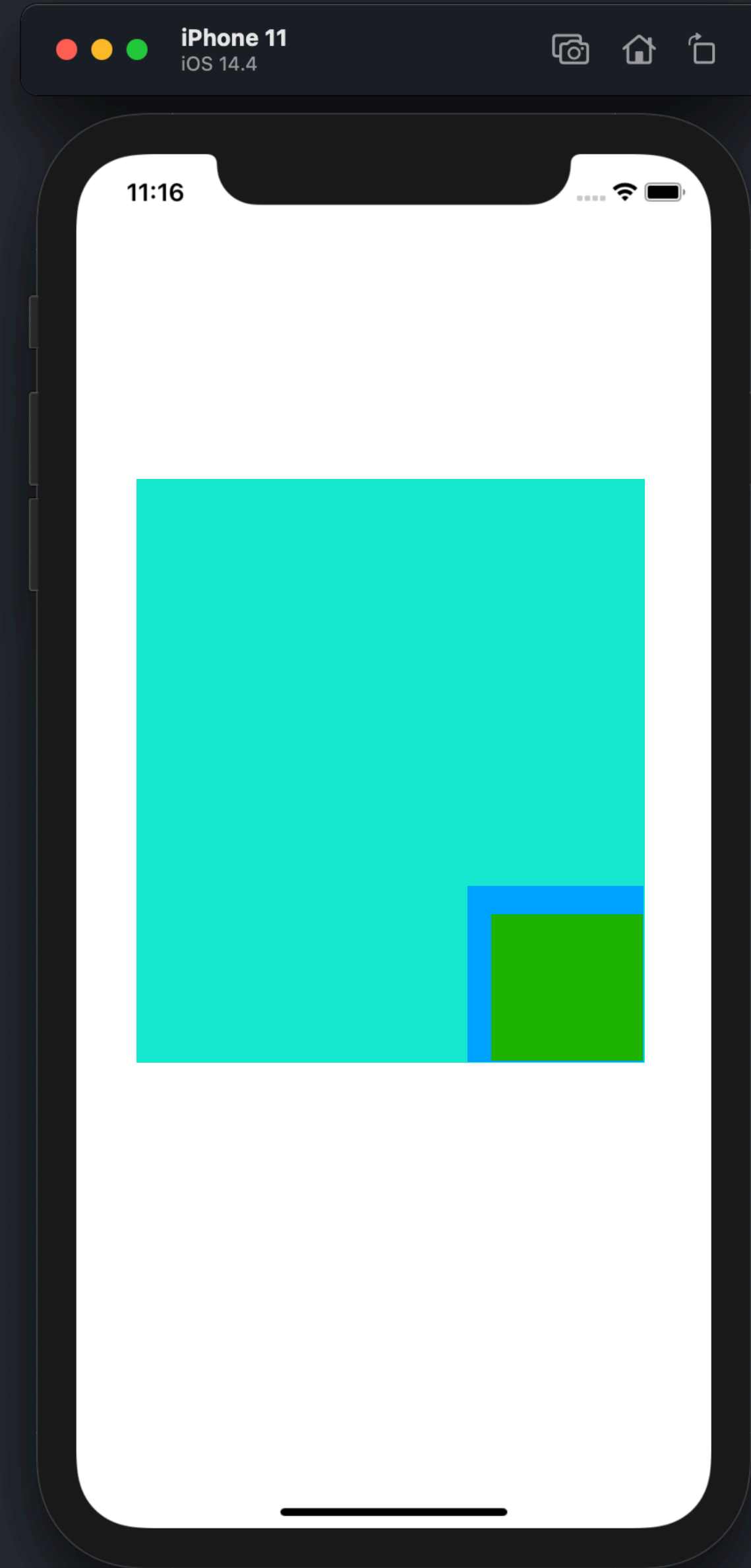
**Alto, ancho, margen,
padding, borde.**

Position

Flex Box

**Dirección, ubicación,
alineamiento, estirar,
encoger, proporciones.**

Position



Relativa

Absoluta

**Absoluta, relativa, top,
left, right, bottom**

Box Object Model

**Alto, ancho, margen,
padding, borde.**

Position

Flex Box

**Dirección, ubicación,
alineamiento, estirar,
encoger, proporciones.**

Mantener una estructura consistente entre diferentes pantallas.

Flex Box

Indica como el elemento y sus hijos deben de ubicarse en el espacio disponible que se es asignado

Stack Navigation

Drawer Navigation

BottomTab Navigation

MaterialTop Navigation

Stack Navigation



A diagram illustrating a navigation stack. It consists of four vertical white rectangles of increasing width, creating a 3D effect. The rectangles are labeled from left to right: 'App', 'Settings', and 'ProfileScreen'. The 'App' label is partially obscured by the 'Settings' rectangle, and the 'Settings' label is partially obscured by the 'ProfileScreen' rectangle. This visualizes a stack where 'ProfileScreen' is the top-most screen, followed by 'Settings', and 'App' is at the bottom.

App Settings ProfileScreen

Stack Navigation



A diagram illustrating a navigation stack. It consists of four vertical white rectangles of increasing width from left to right, creating a 3D effect. The rectangles are labeled from left to right: 'App', 'Settings', and 'ProfileScreen'. The 'App' label is partially obscured by the 'Settings' rectangle, and the 'Settings' label is partially obscured by the 'ProfileScreen' rectangle. This visualizes the concept of a stack where the most recent screen is on top.

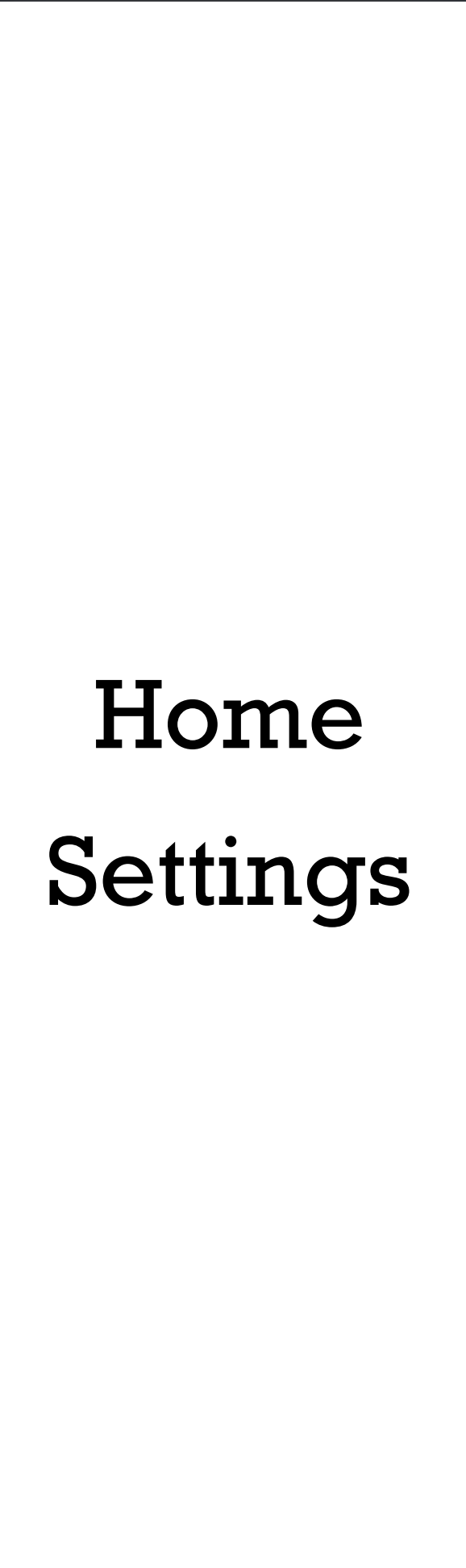
App Settings ProfileScreen



Drawer Screen

Drawer Navigation

Drawer



Drawer Navigation



Stack

Página sola
Stack
Tabs
TopTabs

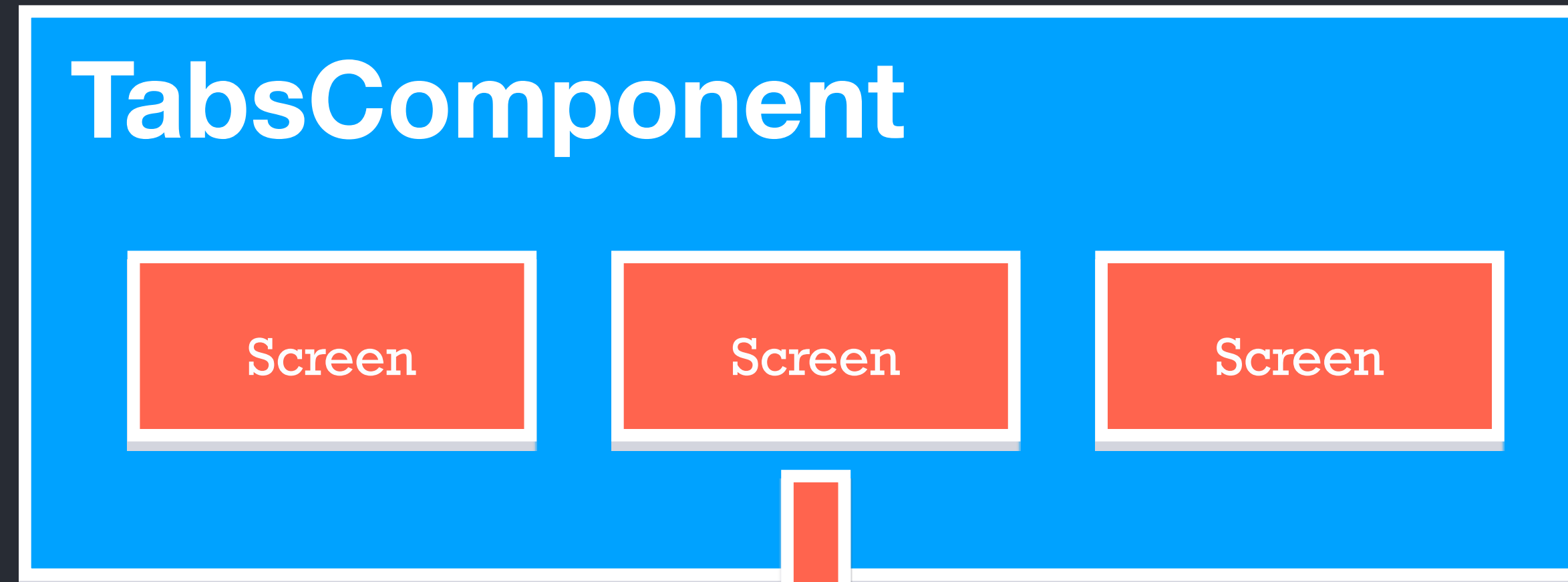
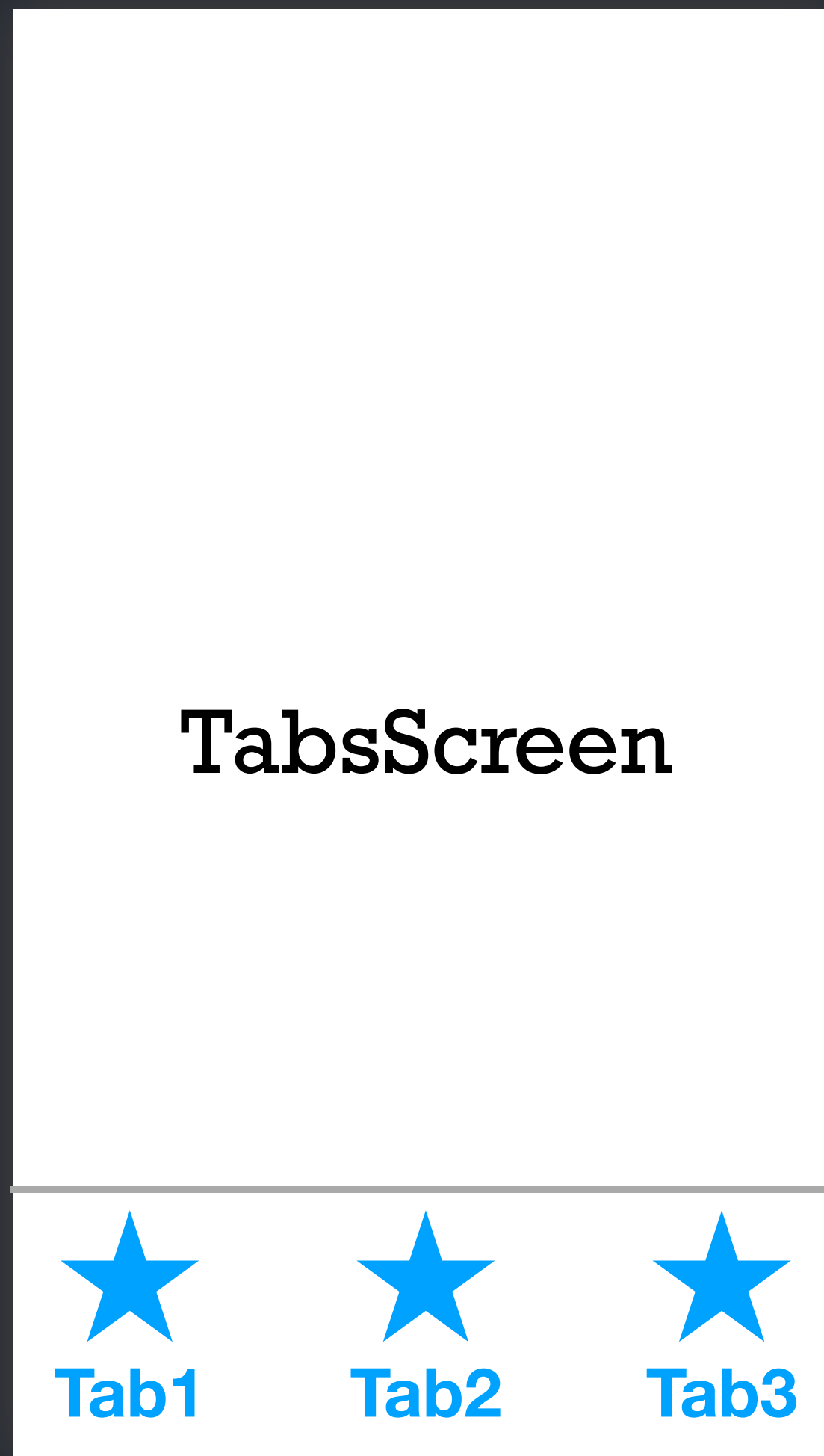
Stack Navigation

Drawer Navigation

BottomTab Navigation

MaterialTop Navigation

BottomTab Navigation



Página sola
Stack
Tabs
TopTabs

