



**Asignatura:** Diseño de interfaces y principios básicos de UI/UX

**Guía de aprendizaje N.º 1**  
20 de Julio de 2024

Managua, 20 de Julio de 2024

## Diseño de interfaces y principios básicos de UI/UX

### I. CONTENIDOS

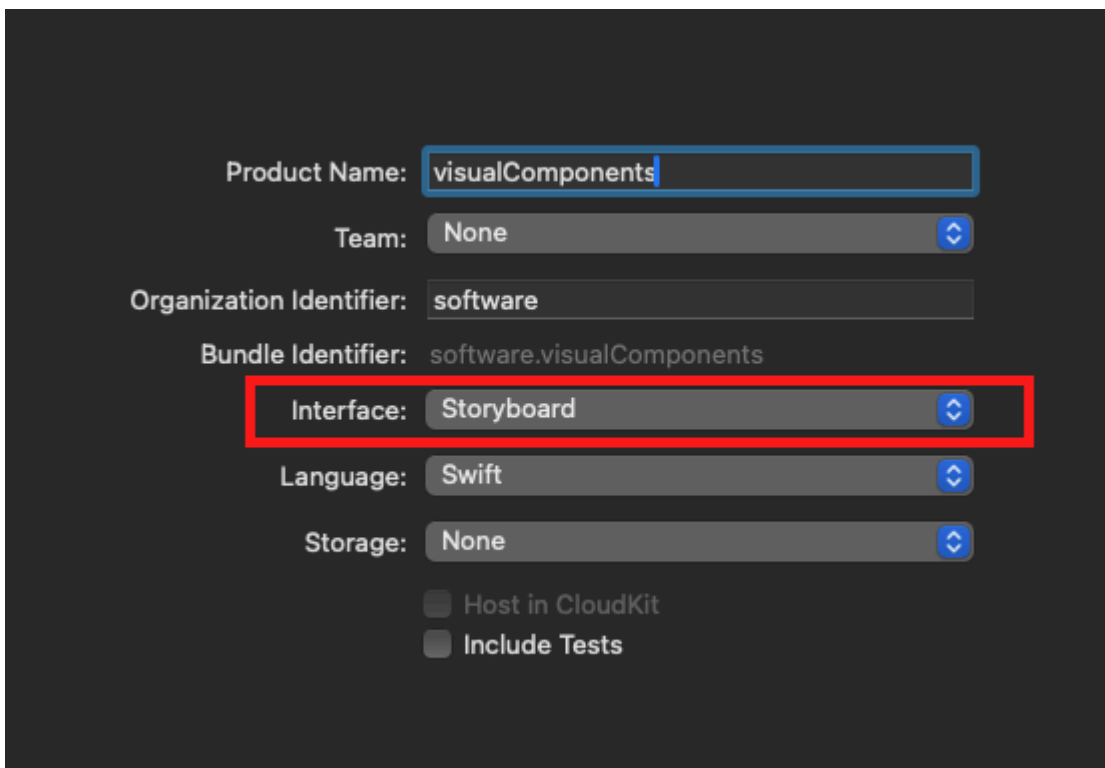
#### Introducción:

En este documento se creará una serie de procedimientos relacionados a los componentes visuales del toolbox de xcode (Label, Button), el objetivo es poder comprender las funciones básicas de los componentes más usados en las aplicaciones.

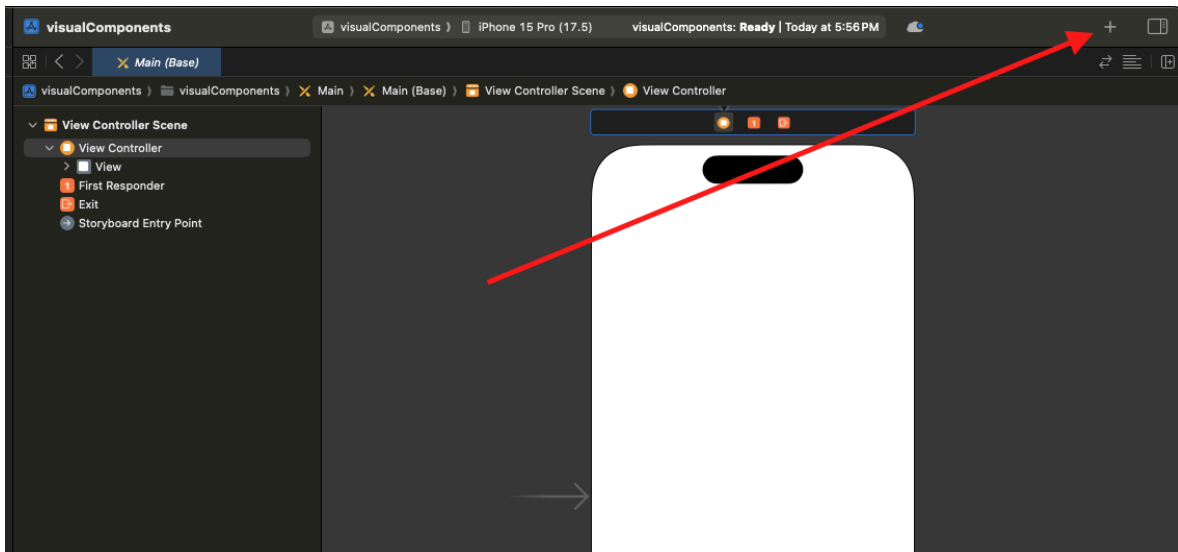
**Crear un nuevo proyecto de Xcode llamado VisualComponents y guardarlo en tu carpeta de proyectos.**

#### Pasos:

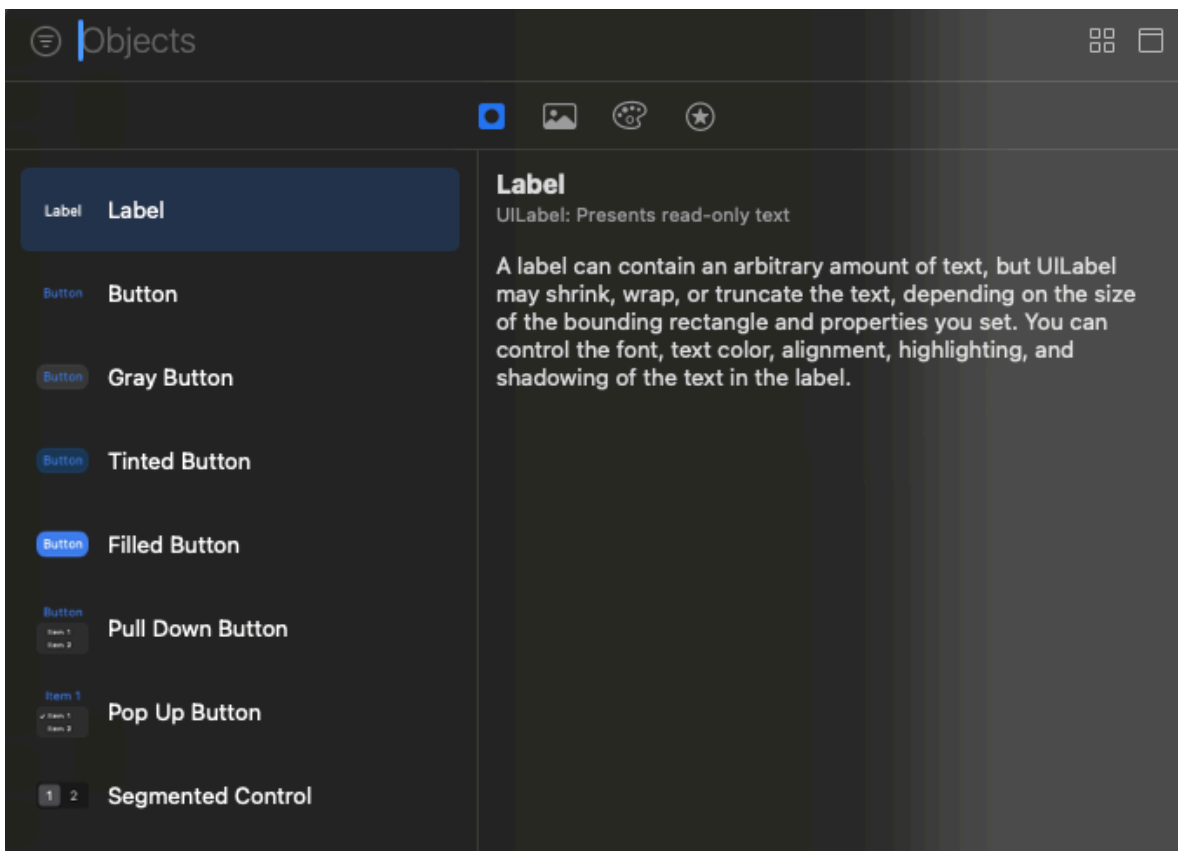
Tener en cuenta la interfaz (Interface) a utilizar, en este caso Storyboard.



Comenzaremos dando un vistazo al toolbox de xcode, para esto dar click en el botón “+” de la parte superior derecha



Al momento de dar click se nos abrirá una caja donde podremos ver cada uno de los componentes visuales con los que cuenta xcode y una breve introducción del componente, como observas puedes ver Label, Button, Segmented Control etc.

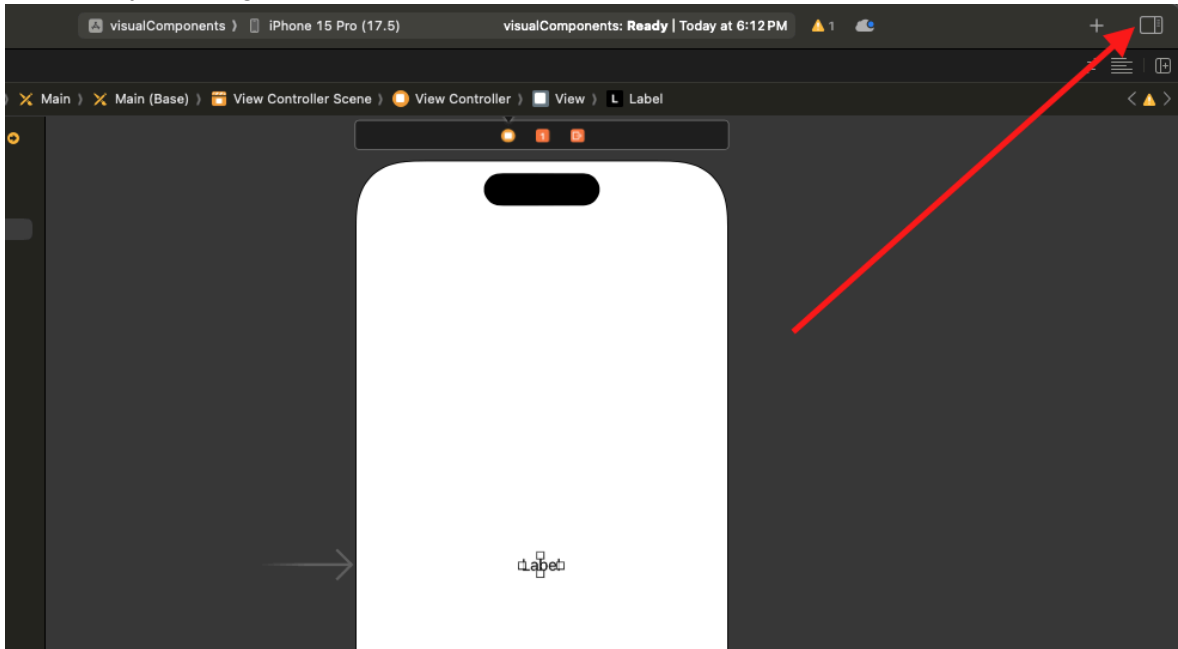


Para insertar Componentes dentro de nuestra interfaz lo único que debemos hacer es arrastrar y soltar, En este ejemplo arrastre el componente Label y lo soltaré hasta la interfaz.

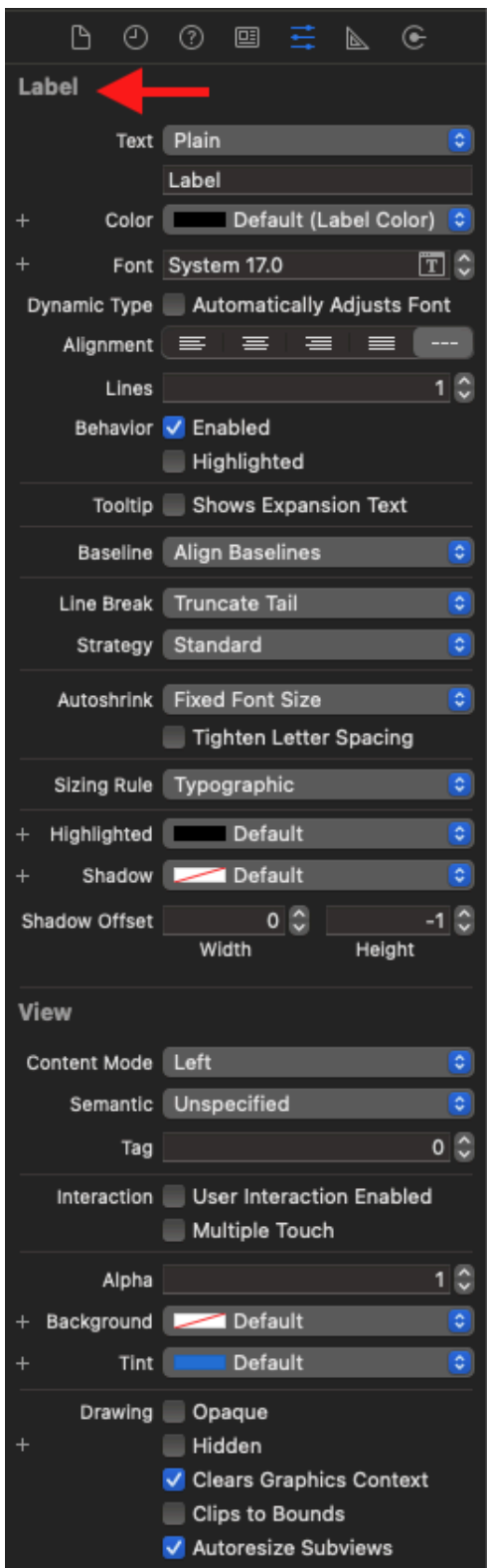


Al momento de tener componentes visuales dentro de nuestra interfaz en este caso nuestro primer Label, ya podemos jugar con diferentes procesos, como darle color, cambiar el nombre, cambiar fuente, tamaño etc.

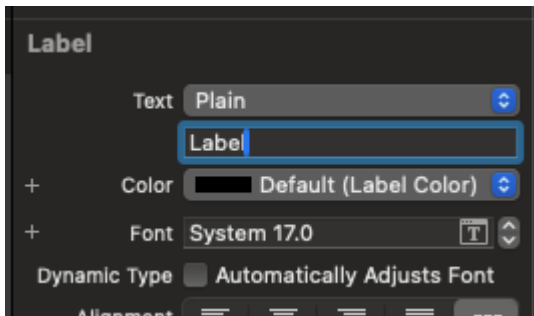
Lo que debemos hacer es seleccionar el label y dar click en el siguiente icono en dado caso que ya lo tengas abierto, omitir este paso.



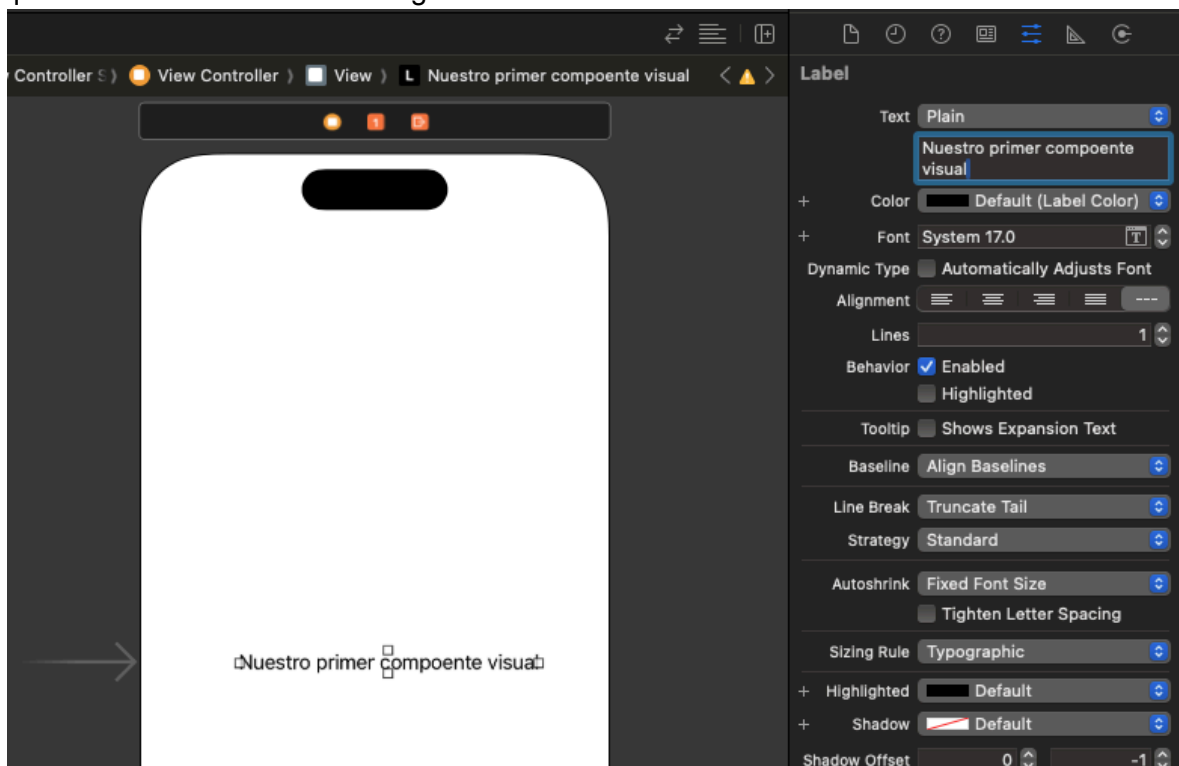
Al dar click, observarás una caja en donde tendremos varias opciones de funcionamiento y verás que al inicio sale el nombre del componente, en este caso Label.



Es momento de realizar algunos ajustes, para empezar cambiaremos el texto de nuestro label a “nuestro primer componente visual”, para eso en la caja donde dice Label, podemos cambiar el nombre.



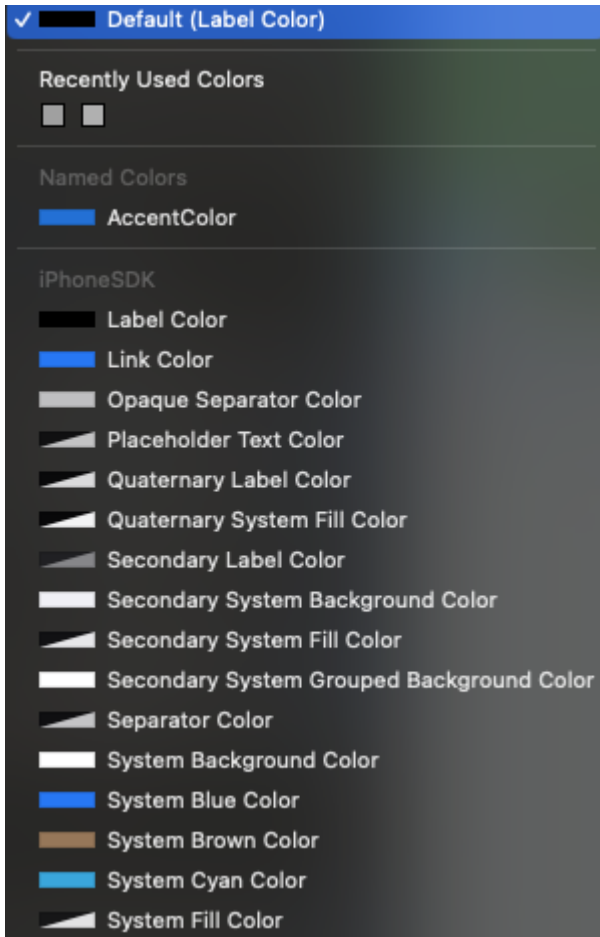
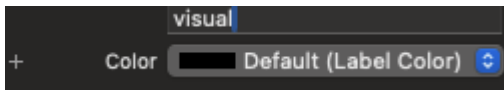
quedando nuestro label de la siguiente manera.



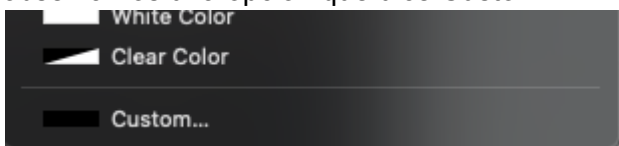
Y listo, hemos creado nuestro primer componente visual.

Vamos a profundizar más, y vamos a cambiar el color, el tamaño de fuente y el tipo de fuente.

Como observamos hay un caja en la que dice Color, damos click y se nos abrirá un panel de colores.

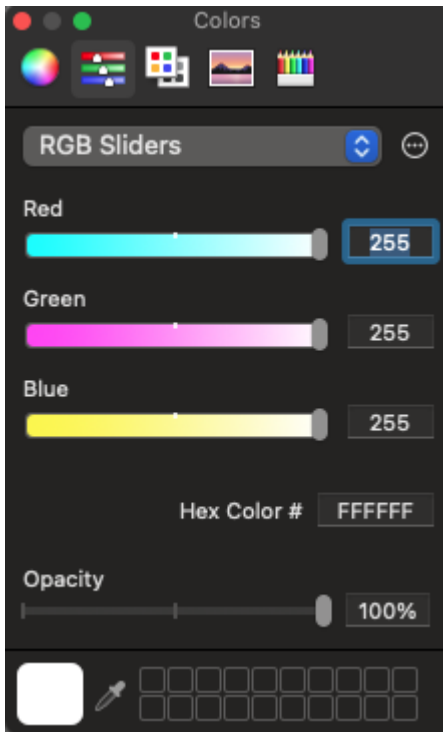


Estos colores que observas están por defecto, pero si nos vamos al final de la lista, observamos una opción que dice Custom





Se nos abrirá una ventana, con más opciones para personalizar el color de nuestro Label

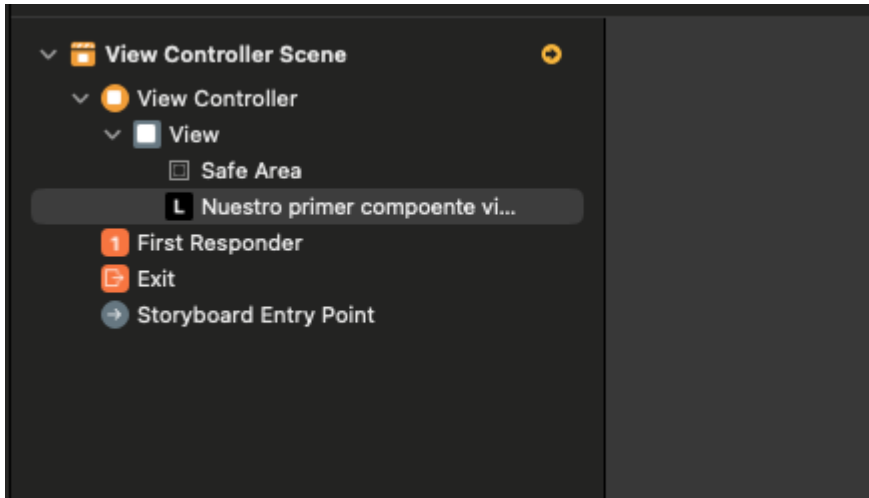


Seleccionamos o armamos el color que queramos, y listo hemos dado color a nuestro Label, en este caso seleccione un color rojo.

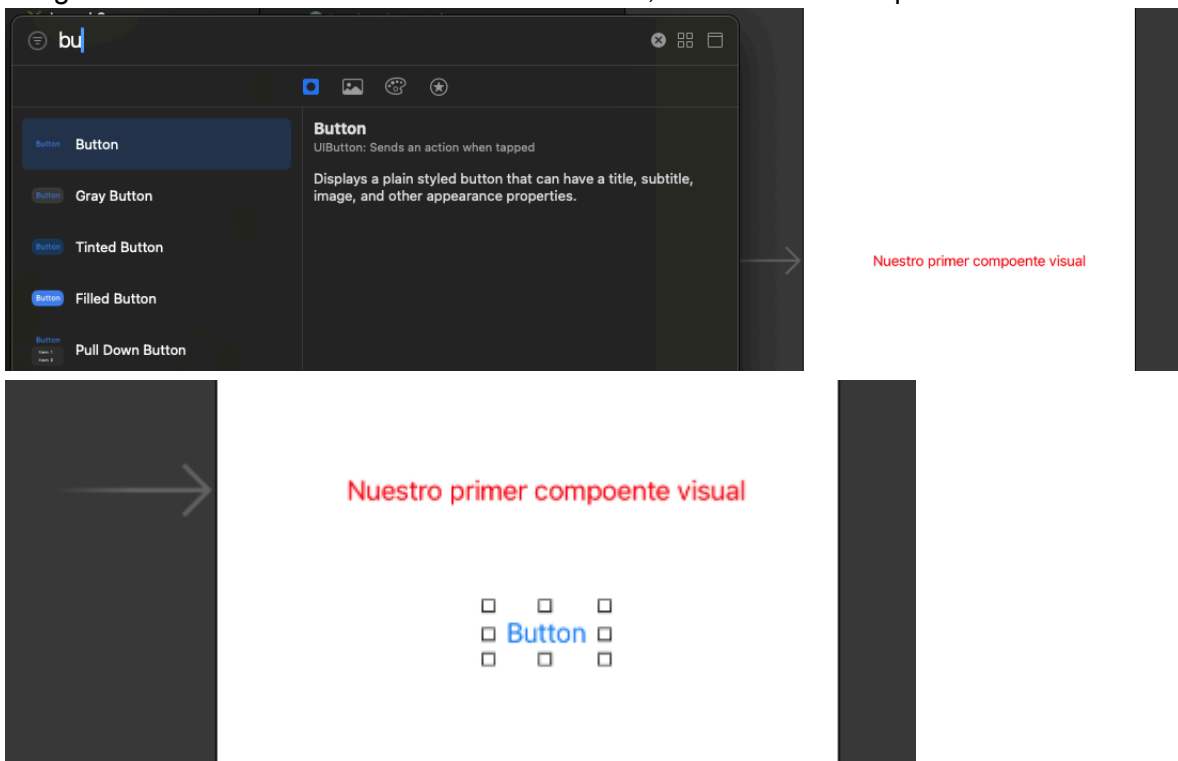


Es momento de hacer algunos ajustes por tu cuenta.  
Intenta cambiar la fuente y el tamaño de este label.

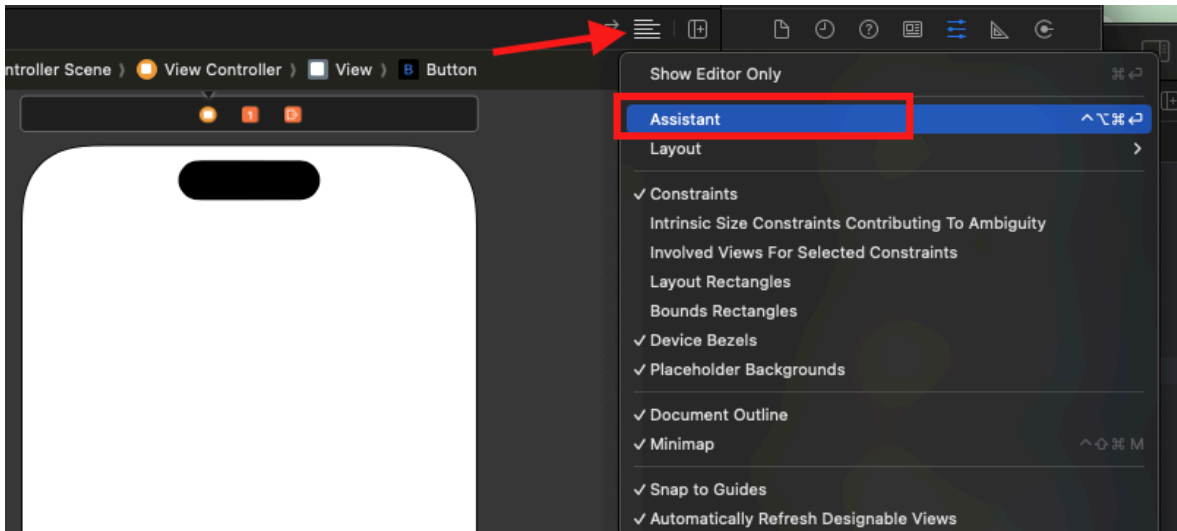
Observa la parte lateral izquierda, esta parte nos va mostrando una jerarquía de los componentes que vayamos añadiendo a nuestra interfaz.



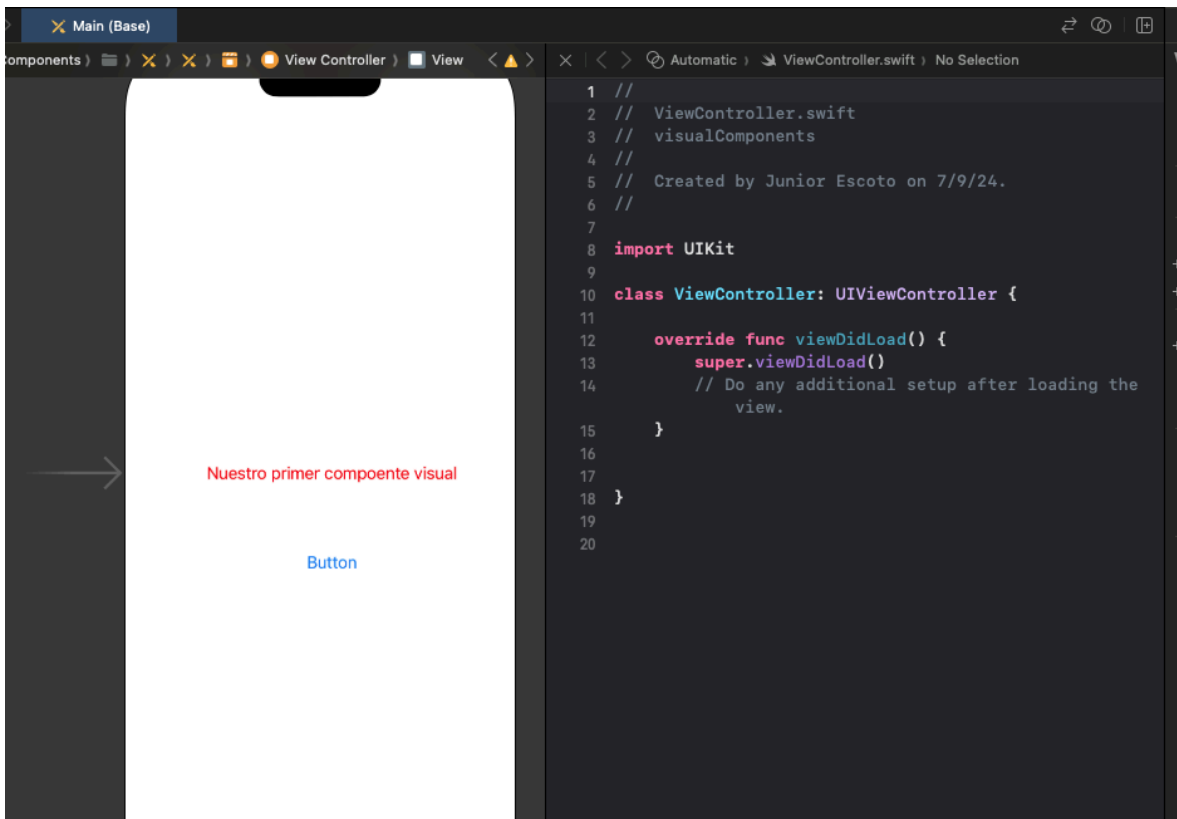
Pongamos otro elemento visual a nuestra interfaz, en este caso busquemos un **“button”**



Pero en este caso vamos a realizar los mismos procedimientos pero a nivel de código. Para empezar debemos enlazar nuestro código a nuestro viewController. click en el siguiente icono mostrado en la imagen, y seleccionamos la opción de **assistant**

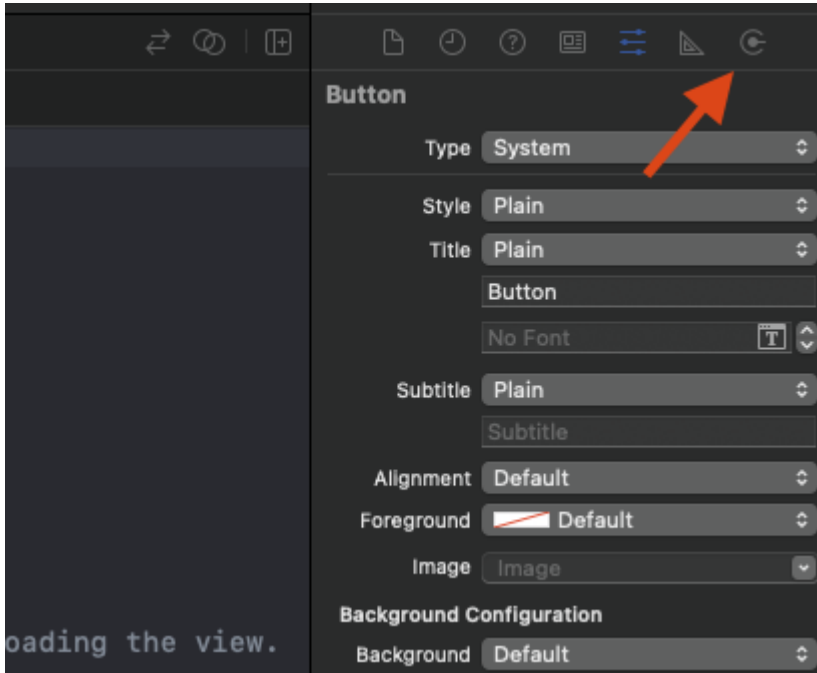


Observaras que nos enlaza y nos muestra el viewController relacionado a nuestra interfaz.

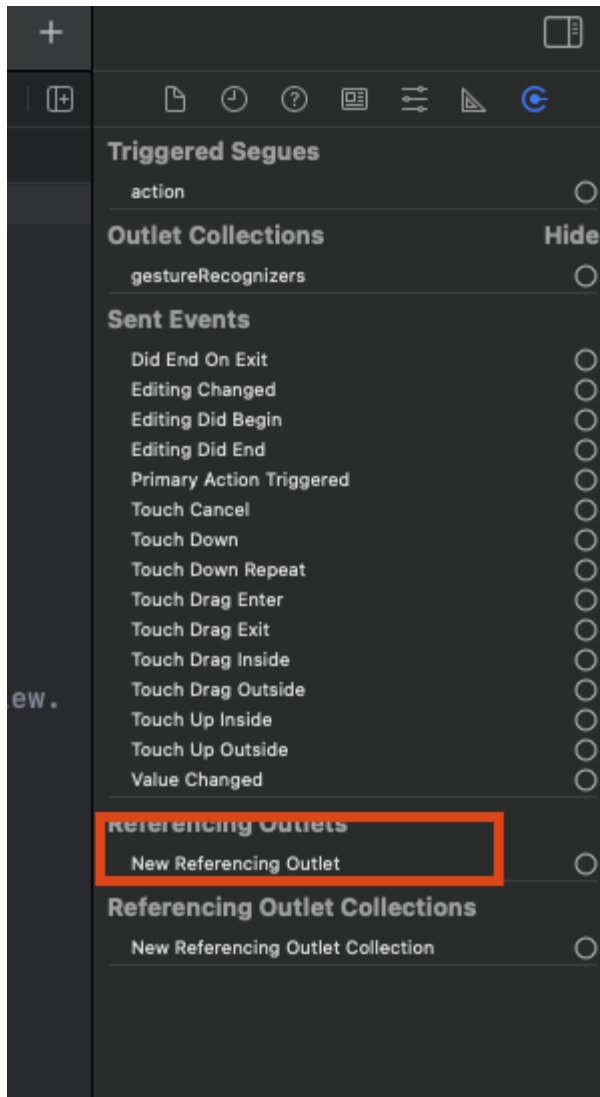


Lo siguiente sera enlazar nuestro botton a nuestra interfaz, lo que en xcode se llama **outlet**

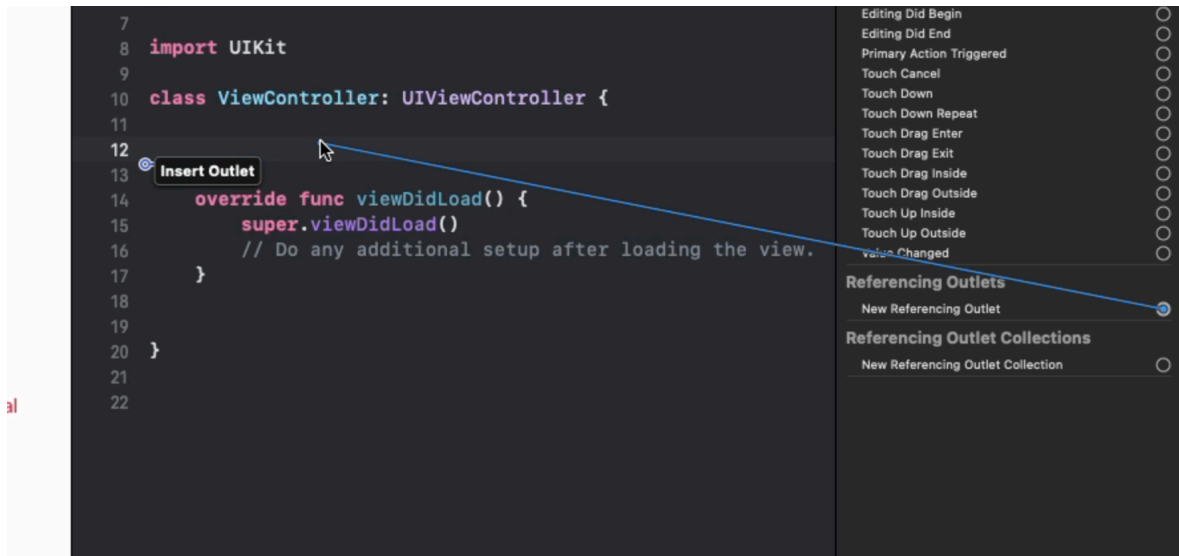
En nuestra caja de herramientas de propiedades, seleccionamos la ultima opcion, igual como en la imagen.



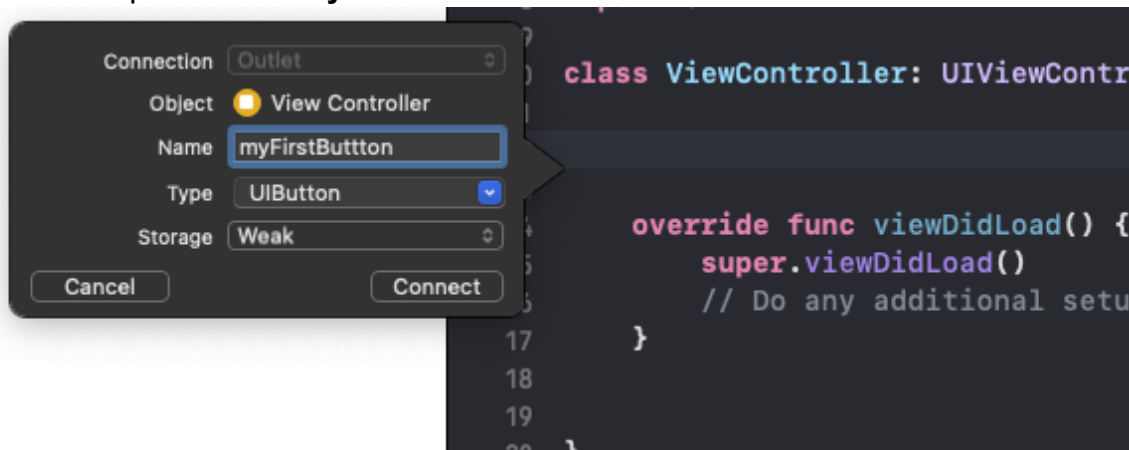
Al dar click, se nos abrirá otra serie de propiedades y funcionalidades, el cual nos interesa la opción de **New Referencing Outlet** .



lo que vamos hacer ahora es dar un click sostenido, dentro del circulo y arrastras hasta nuestro viewController.



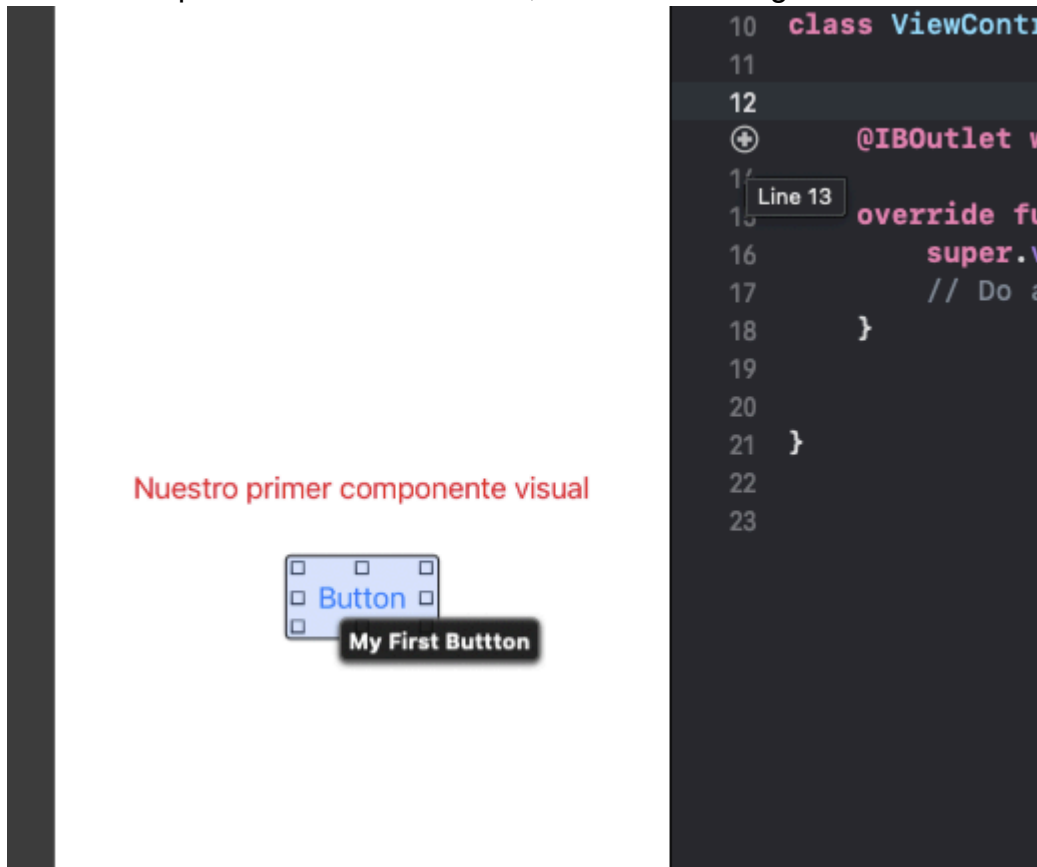
Luego se nos abrirá una cajita, la cual nos pedirá el nombre de nuestro botón, es este caso le pondremos **myFirstButton**.



y damos en Connect.

listo, hemos conectado nuestro boton de nuestra interfaz en nuestro view controller, si pasas el mause sobre el circulo en el codigo veraz como se resaltan

tanto el componente de nuestra vista, como en el código.



Ahora vamos hacer algunos ajustes, pero a nivel de código

Escribimos el siguiente código dentro de nuestra función **viewDidLoad**

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view.

    // Buttons
    myFirstButton.setTitle("myFirstButton", for: .normal)
    myFirstButton.backgroundColor = .blue
    myFirstButton.setTitleColor(.white, for: .normal)
}
```

como observamos he hecho referencia a mi botón llamado **myFirstButton**, y he establecido un título con la función **setTitle**, el **for** que observas, es prácticamente si quieres que el botón tenga un estilo, en este caso le puse el por defecto, que sería normal, tu puedes probar con las demás opciones para ver los demás estilos.

Luego he establecido un `backgroundColor`, que sería un color de fondo a mi botón y por último un color a mi título, con la función `setTitleColor` en blanco. y este sería el resultado, corre nuevamente el simulador.

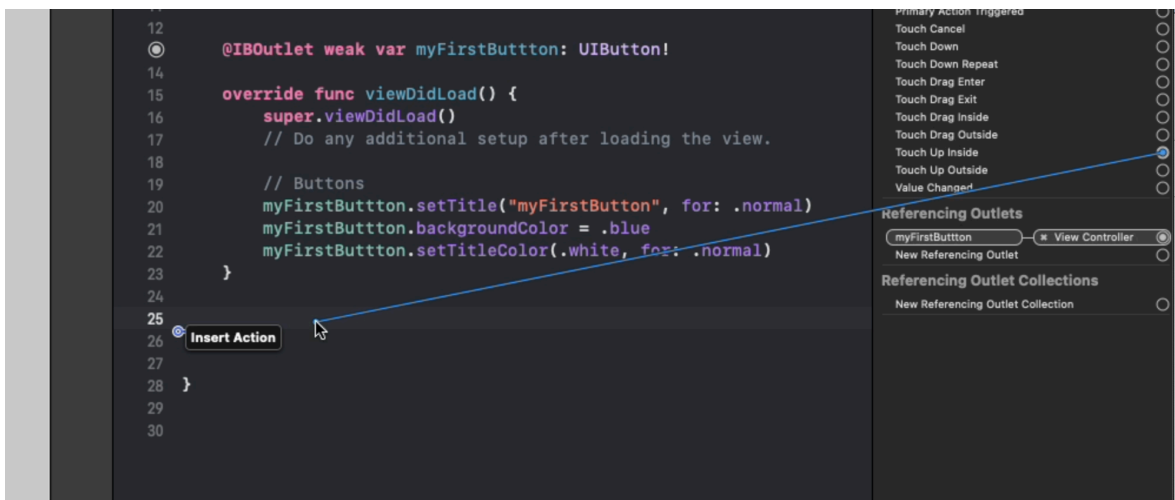
### Nuestro primer componente visual



No olvides estirar el botón desde la interfaz para que el texto se logre ajustar al contenido y listo, hemos conectado y ajustado un botón de la interfaz en nuestro propio `viewController`.

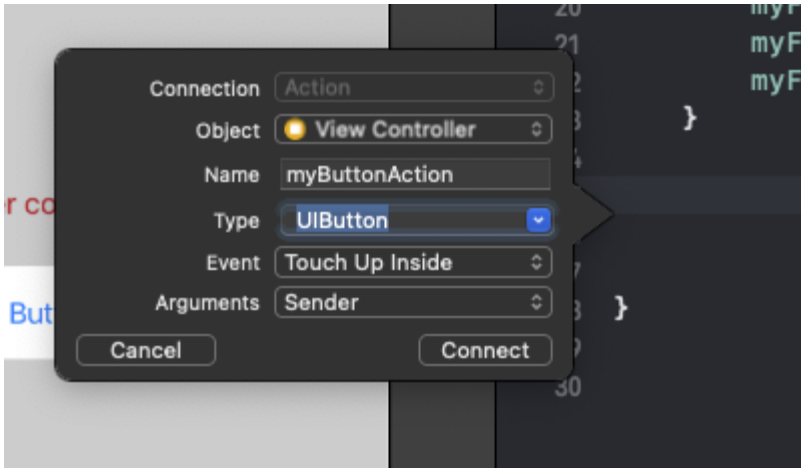
Ahora vamos a programar una acción a nuestro botón.

nos devolvemos a nuestra caja de herramientas donde habíamos enlazado el outlet, pero ahora será la propiedad **Touch up Inside**, igual click al círculo y arrastramos a nuestro controller, pero observa aquí, cómo es una acción a nuestro botón, por convención este lo escribimos por debajo de nuestro **viewDidLoad**.



Igualmente se nos abra una caja, en la cual pondremos el nombre de nuestra función. en este caso le pondré **myButtonAction**





Y damos Connect.

juguemos con un poco de logica, lo que quiero es que cada vez que presione mi boton este cambie de color.

```
25
26  @IBAction func myButtonAction(_ sender: UIButton) {
27
28      if myFirstButttton.backgroundColor == .blue {
29          myFirstButttton.backgroundColor = .green
30      } else {
31          myFirstButttton.backgroundColor = .blue
32      }
33  }
```

Como observas algo sencillo.. si el color de mi boton es .blue, cambialo a .green, de lo contrario cuando este esté en .green, regresalo a .blue. vuelve a compilar y observa los cambios.

Nuestro primer componente visual

myFirstButton

Reto: Vuelve la lógica de if else a una lógica ternaria.

## Ejercicios

**Descripción:** Los estudiantes realizarán una serie de ejercicios prácticos que cubrirán todos los temas tratados en la clase. Los ejercicios permitirán a los estudiantes poner en práctica lo aprendido y recibir retroalimentación inmediata sobre su comprensión.

## II. LOGROS DE APRENDIZAJES

Al finalizar el encuentro, los estudiantes serán capaces de:

- Conocer los componentes visuales del toolbox de xcode Label, Button
- Poder implementar componentes visuales, detallando colores y funcionamiento.
- Identificar la caja de herramientas, storyboard y menú de implementación...

### III. ORIENTACIONES METODOLÓGICAS

Para alcanzar los logros de aprendizaje, luego de la evaluación trabajaremos con las siguientes estrategias y recursos:



#### **Actividad de Aprendizaje No.1**

**Modalidad:** Individual

**Tiempo aproximado:** 10 minutos

**Periodo de realización:** 20 de Julio, en el aula de clases

**Puntaje:** Ninguno

**Descripción de la actividad:** Crear un label y un botón en la interfaz visual, crear los respectivos outlet a cada uno. Cambia el color del label a un color amarillo, Desde la interfaz visual crea una acción al botón, para que el texto(text) y el color del label(textColor) cambie.