

# Píldoras SwiftUI

## GeometryReader



### Conceptos simples

Lo que necesitas saber en una primera instancia para luego poder manejar la complejidad de conceptos más avanzados.

### Nivel principiante

Explicaciones de mínima complejidad para poner foco en lo que importa.

### Código de prueba

Código corto y simple para que pongas a prueba ideas y conceptos.

### Casos reales

Situaciones de aplicación en casos reales para que visualices el ejemplo como una herramienta lista para usar.

### Explicaciones claras

Explicaciones que explican, no que complican..

### Preguntas de entrevistas

Preguntas y respuestas sobre el tópico que usualmente aparecen en entrevistas laborales.

El conocimiento en programación es como la construcción de un edificio: la solidez de toda la estructura depende de la calidad de cada ladrillo. Cada unidad de conocimiento debe ser clara, sólida y comprensible por sí misma, independiente de las demás. Con el tiempo, estos bloques se consolidan y se conectan creativamente con otros, formando una estructura robusta y armoniosa.

GeometryReader es una herramienta avanzada en SwiftUI que suele abordarse en los capítulos finales de los libros, pero su uso es valioso desde los primeros pasos en la creación de una interfaz. GeometryReader puede ser tan sencillo como proporcionar información sobre el contexto gráfico para posicionar un componente o tan sofisticado como para controlar animaciones complejas. Si estás empezando a construir tu "edificio" en Swift, este es el momento ideal para incorporar y perfeccionar este pequeño pero poderoso ladrillo en tu estructura.

## ¿Qué es GeometryReader?

GeometryReader es una vista especial en SwiftUI que proporciona información sobre el tamaño y la posición del espacio disponible en la vista a la cual envuelve. Al usarla, accedes a un objeto GeometryProxy, que contiene datos sobre el tamaño y las coordenadas de la vista. Esto permite ajustar el diseño de forma dinámica según el espacio disponible, similar a cómo AutoLayout funcionaba en UIKit.

## ¿Cuándo y por qué apareció?

GeometryReader fue introducido en 2019 con el lanzamiento de SwiftUI en iOS 13. Su inclusión se debió a la necesidad de contar con herramientas de diseño adaptativo en SwiftUI que permitieran a los desarrolladores crear interfaces de usuario fluidas y adaptables a diferentes tamaños de pantalla. Antes de SwiftUI, el diseño adaptativo en UIKit se lograba a través de Auto Layout y otras herramientas visuales en Interface Builder. Sin embargo, con SwiftUI y su enfoque declarativo, era fundamental contar con una forma programática de acceder al espacio de diseño y ajustar las vistas dinámicamente.

La introducción de GeometryReader ayudó a resolver estos problemas al proporcionar una manera eficiente de leer el tamaño y la posición de una vista en tiempo real, permitiendo a los desarrolladores responder a cambios en el entorno y creando así interfaces más responsivas.

## ¿Para qué sirve?

GeometryReader permite construir vistas que se ajustan automáticamente a diferentes tamaños de pantalla. Es útil para escribir lógica que reorganice o redimensione elementos cuando cambia la orientación del dispositivo (como al pasar de vertical a horizontal en un iPhone) o cuando una app se ejecuta en dispositivos de distintos tamaños (por ejemplo, de un iPhone a un iPad). Esto hace que las vistas sean más adaptativas y respondan de forma fluida a distintos espacios de visualización. Si no cuentas con la ayuda de GeometryReader, esta tarea puede ser realmente desafiante de implementar y compleja de mantener cuando Apple libera nuevos dispositivos con espacios de visualización diferentes.

## ¿Cuándo debe utilizarse?

GeometryReader debe usarse cuando necesitas que una vista o varios elementos dentro de ella se adapten automáticamente al tamaño y posición disponibles en pantalla. Esto es especialmente útil en situaciones como:

- Adaptación a diferentes tamaños de pantalla: cuando una misma interfaz debe verse bien tanto en un iPhone como en un iPad, o en cualquier dispositivo con diferentes dimensiones.
- Reaccionar a cambios de orientación: al rotar el dispositivo de vertical a horizontal, GeometryReader permite ajustar el diseño para aprovechar mejor el espacio.
- Distribuir elementos en función del espacio disponible: si quieres posicionar o dimensionar elementos en relación a la pantalla o a su contenedor, GeometryReader te da los datos para hacerlo.

## ¿Contraindicaciones?

Sí, GeometryReader puede ser costoso en términos de cálculo computacional, lo que implica un mayor consumo de batería. Intenta estructurar la funcionalidad de tu vista sin depender de GeometryReader y úsalo solo cuando sea realmente necesario. Ten en cuenta que una vista con varios GeometryReader podría ralentizar el rendimiento en algunos dispositivos.

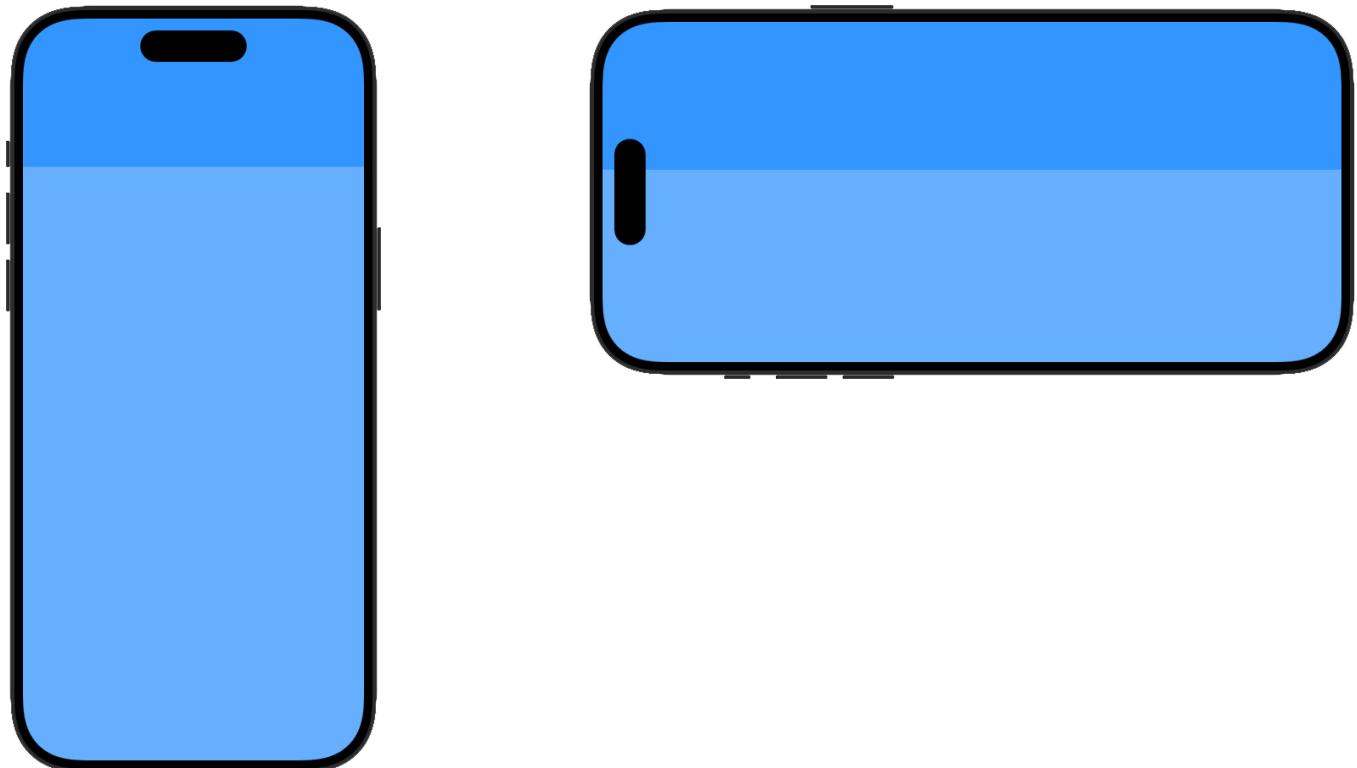
Veamos un caso de uso simple para ver como funciona

Supón que defines una cabecera en tu vista que siempre debe ocupar el 30% de la altura total, sin importar el dispositivo o si lo giras a modo horizontal.

Una primera aproximación seria la siguiente:

```
struct View001: View {
    var body: some View {
        VStack(spacing: 0) {
            VStack {
                Rectangle()
                    .fill(Color.blue.opacity(0.8))
                    .frame(height: UIScreen.main.bounds.height * 0.20)
            }
            VStack {
                Rectangle()
                    .fill(Color.blue.opacity(0.6))
            }
        }
        .ignoresSafeArea()
    }
}
```

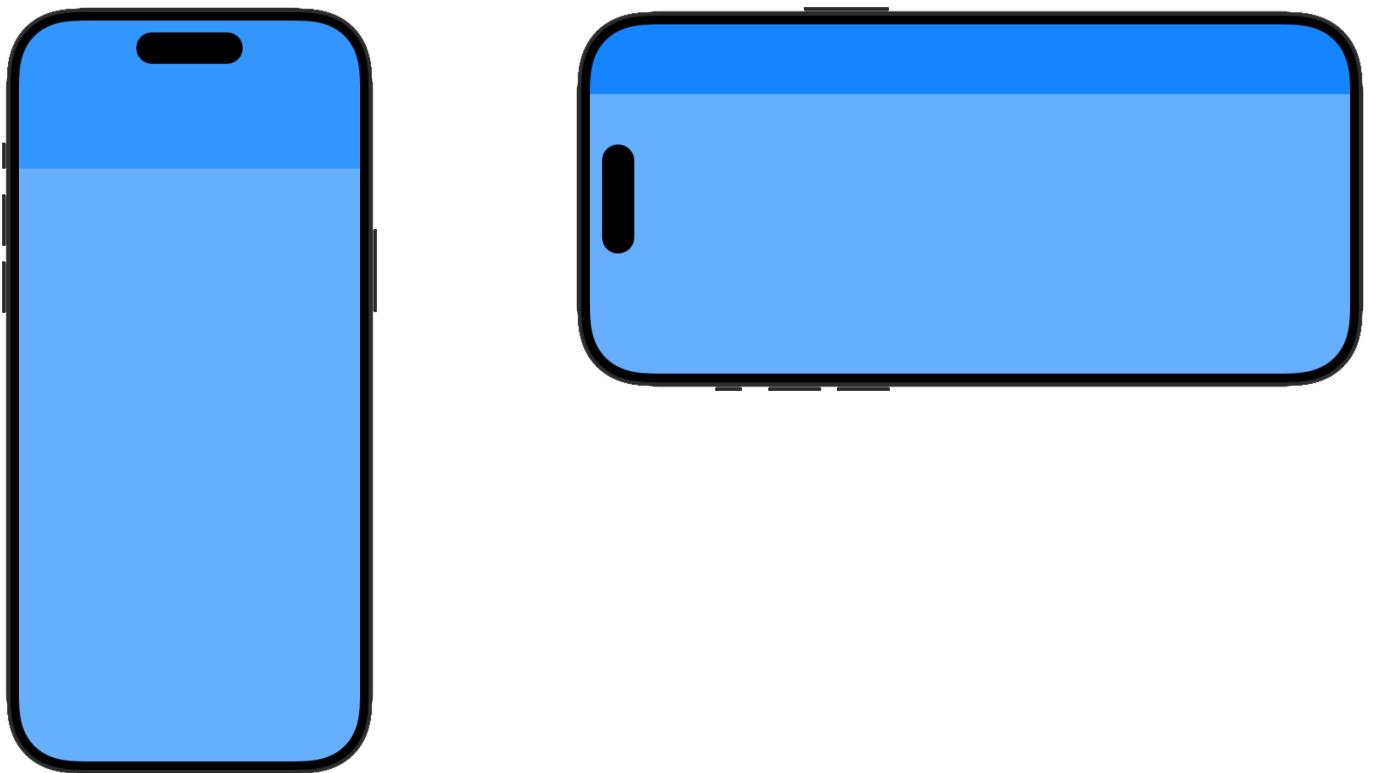
El resultado con pantalla vertical es correcto, sin embargo el resultado con pantalla horizontal no es el esperado. Esto curre porque la altura referenciada en el frame toma el alto de la "pantalla física" que es el mismo ya sea que esta esté en posición vertical u horizontal.



Para solucionar este problema podemos recurrir a envolver nuestros stacks en un GeometryReader para luego utilizar la lectura de geometría para referirnos a la altura de la vista en lugar de la altura de la pantalla, obteniendo el resultado esperado.

```
struct View001: View {
    var body: some View {
        VStack(spacing: 0) {
            GeometryReader { geometry in
                VStack {
                    Rectangle()
                        .fill(Color.blue.opacity(0.8))
                        .frame(height: geometry.size.height * 0.20)
                }

                VStack {
                    Rectangle()
                        .fill(Color.blue.opacity(0.6))
                }
            }
            .ignoresSafeArea()
        }
    }
}
```



GeometryReader es una herramienta clave para crear diseños responsivos en SwiftUI. Muchas de las preguntas que planteamos al inicio podrían surgir en pruebas técnicas de entrevistas laborales. No necesitas ser un experto en GeometryReader, pero conocer su existencia y saber explicar su funcionamiento básico añade un gran valor como desarrollador y facilita profundizar en el tema cuando llegue el momento.

Gracias por descargar este contenido. Si te ha resultado útil, sígueme en LinkedIn para más píldoras SwiftUI.

Código de ejemplo: <https://github.com/edd709/P20241105151449.git>