

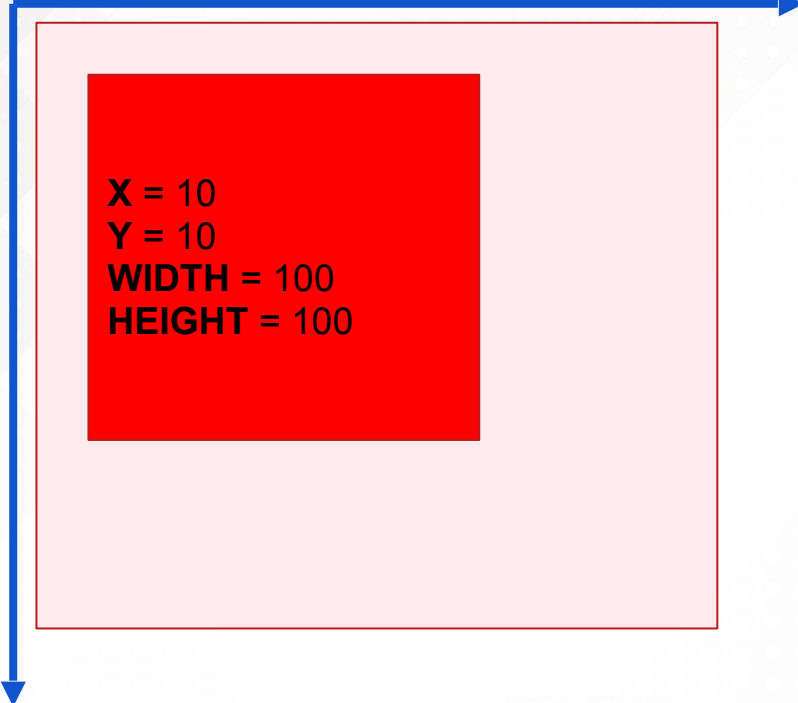
FRAMES

- **CGPoint** → Tem atributos X e Y, ambos numéricos
- **CGSize** → Tem atributos WIDTH e HEIGHT, ambos numéricos
- **CGRect** → Tem atributos ORIGIN (tipo CGPoint) e SIZE (tipo CGSize)

UIView: Atributos

O **frame** de uma visualização é do tipo CGRect:

- ORIGIN (coordenadas de origem) = (X , Y)
- SIZE (tamanho) = (Width , Height)



X = 10
Y = 10
WIDTH = 100
HEIGHT = 100

UIView: Atributos

The screenshot displays the Xcode interface with the 'UIView' Attributes Inspector open. The interface is divided into three main sections: a left sidebar, a central canvas, and a right-hand inspector panel.

Left Sidebar (View Controller Scene):

- View Controller Scene
 - View Controller
 - Top Layout Guide
 - Bottom Layout G...
 - View
 - Green View (selected)
 - First Responder
 - Exit
 - Storyboard Entry Poi...

Central Canvas:

A green rectangular view is shown on the canvas, representing the 'Green View' selected in the sidebar. It has a yellow circle icon, a red cube icon, and a red square icon in the top right corner, and a battery icon in the bottom right corner.

Right Panel (View Attributes Inspector):

View

Show **Frame Rectangle**

20 20
X Y

175 175
Width Height

Arrange **Position View**

Autoresizing

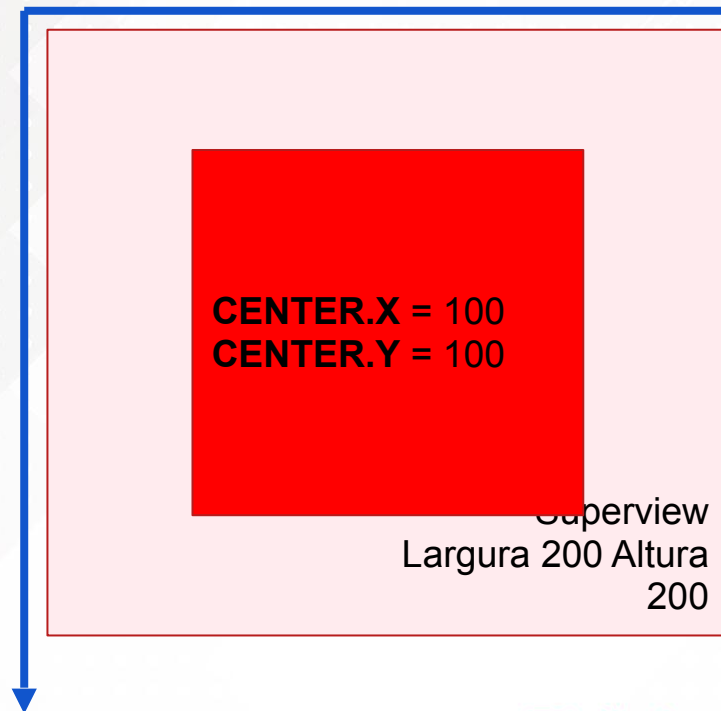
Layout Margins **Default**

- ☐ Preserve Superview Margins
- ☐ Follow Readable Width

UIView: Atributos

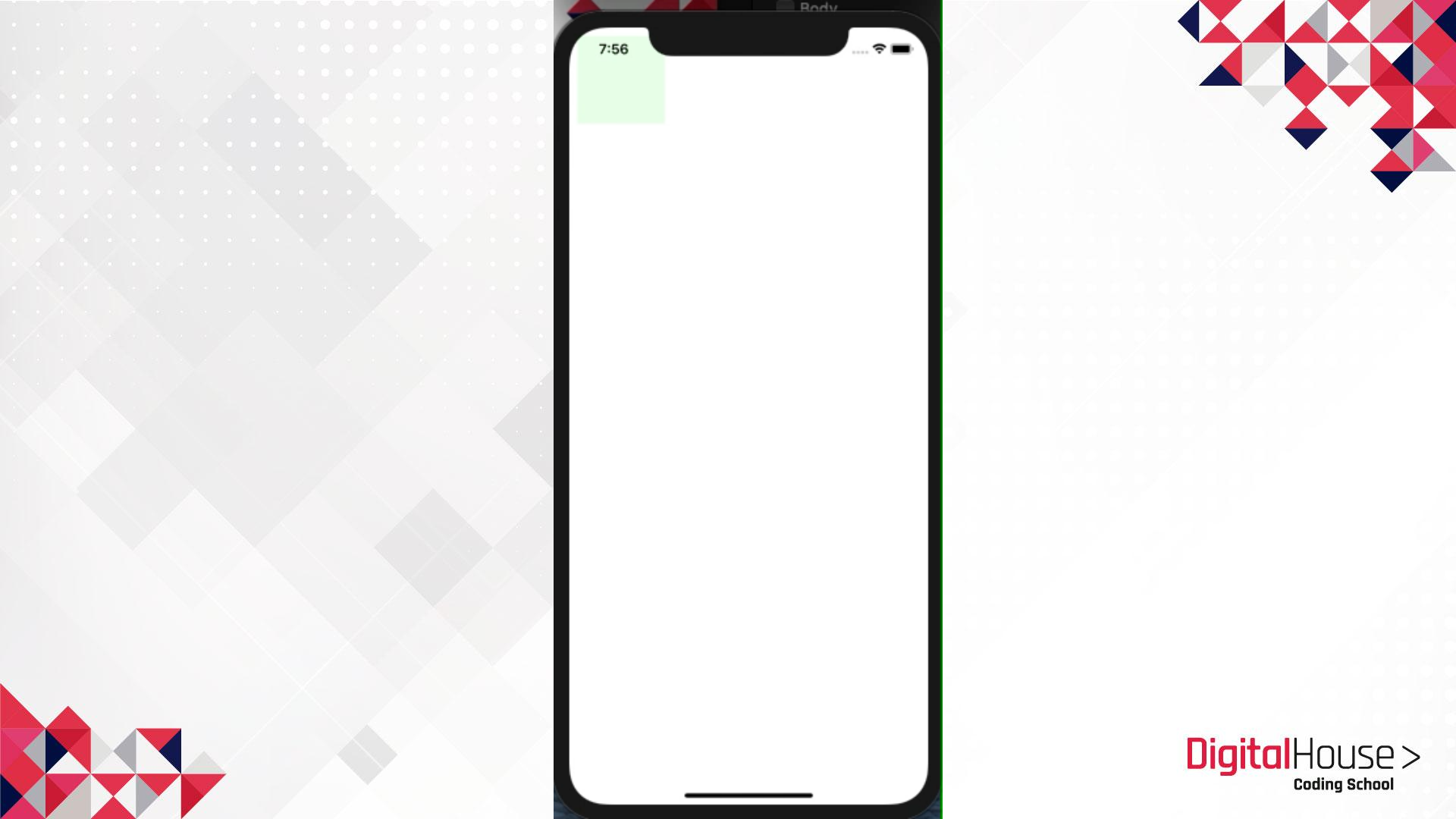
O **center** de uma VIEW é do tipo CGPoint:

- X
- Y



Animation

```
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        addViewAndAnimate()  
    }  
  
    func addViewAndAnimate() {  
        let newView = UIView(frame: CGRect(x: 10, y: 10, width: 100, height: 100))  
        newView.backgroundColor = .green  
        view.addSubview(newView)  
        UIView.animate(withDuration: 3) {  
            newView.frame = self.view.frame  
        }  
    }  
}
```



7:56



Links

Documentação - UIView

Extra animação - Animação

Exercício

Exercício 1:
Crie um app que:

Tenha uma view 50x50 x:0 y:0 na view principal.

Ao dar um tap nessa view, deve adicionar uma nova view do mesmo tamanho abaixo da que recebeu o tap. A nova view deve ter um tap que tenha o mesmo comportamento.

Quando acabar o espaço em tela, deve adicionar uma view no centro da view principal que se anime até ficar do tamanho da tela toda.

Quando terminar a animação ela deve ficar branca. Ao ficar branca, você deve remover todas as subviews da view principal e adicionar novamente a primeira view para começar o ciclo novamente.