

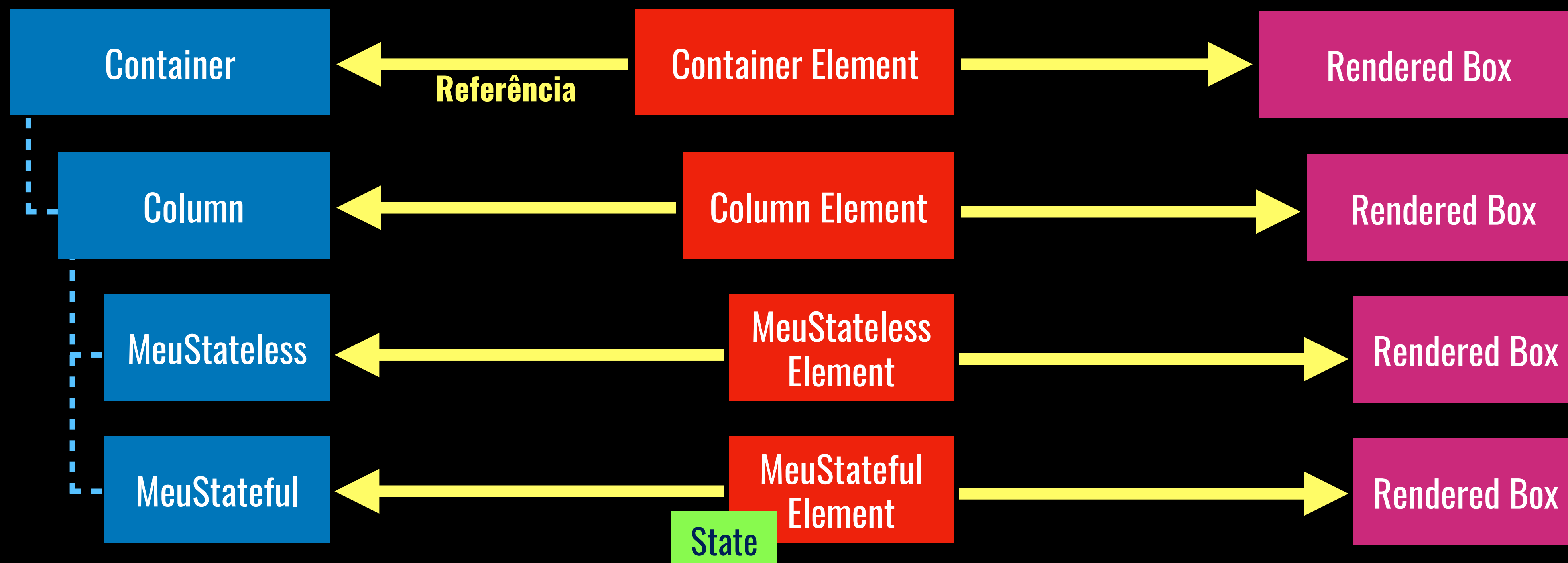
# Árvores do Flutter

Flutter & Widgets Internamente





# Árvore de Widgets & Árvore Elementos



Select Widget Mode Refresh Tree

Slow Animations Debug Paint Paint Baselines Repaint Rainbow Debug Banner Platform: iOS



Expand all Collapse to selected

**Column**  
direction: vertical  
mainAxisAlignment: start  
mainAxisSize: max  
crossAxisAlignment: center  
textDirection: null  
verticalDirection: down  
textBaseline: null  
renderObject: RenderFlex#186e6 relayoutBoundary=up2

**Container**  
null  
padding: null  
bg: null  
fg: null  
constraints: BoxConstraints(0.0<=w<=Infinity, h=27.4)  
margin: null

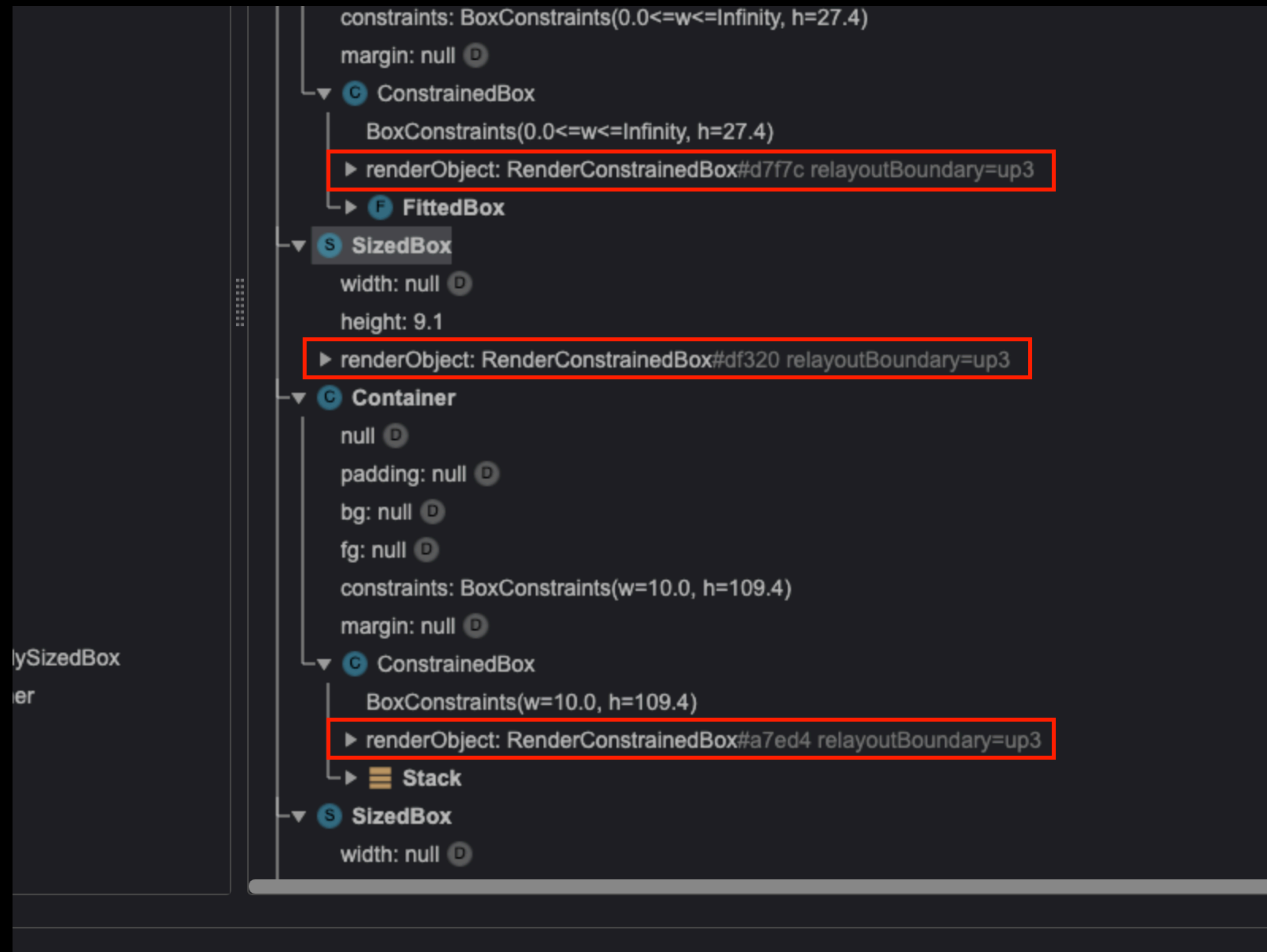
**ConstrainedBox**  
BoxConstraints(0.0<=w<=Infinity, h=27.4)  
renderObject: RenderConstrainedBox#d7f7c relayoutBoundary=up3

**FittedBox**

**SizedBox**  
width: null  
height: 9.1  
renderObject: RenderConstrainedBox#df320 relayoutBoundary=up3

**Container**  
null  
padding: null  
bg: null  
fg: null  
constraints: BoxConstraints(w=10.0, h=109.4)







# Quando o Método **Build** é Chamado?

```
102 @override
103 Widget build(BuildContext context) {
104     final mediaQuery = MediaQuery.of(context);
105     bool isLandscape = mediaQuery.orientation == Orientation.landscape;
106
107     final iconList = Platform.isIOS ? CupertinoIcons.refresh : Icons.list;
108     final chartList =
109         Platform.isIOS ? CupertinoIcons.refresh : Icons.show_chart;
110
111     final actions = <Widget>[
112         if (isLandscape)
113             _getIconButton(
114                 _showChart ? iconList : chartList,
115                 () {
116                     setState(() {
117                         _showChart = !_showChart;
118                     });
119                 }
120             )
121     ];
```







# setState $\Rightarrow$ build

```
_removeTransaction(String id) {  
  setState(() {  
    _transactions.removeWhere((tr) => tr.id == id);  
  });  
}
```

Marcar o objeto como “sujo”





# MediaQuery.of ⇒ build

```
// Padding
C MediaQueryData of(BuildContext context, {bool nullOk = false})
i package:flutter/src/widgets/media_query.dart
e The data from the closest instance of this class that encloses the given context.
T You can use this function to query the size and orientation of the screen. When that
l information changes, your widget will be scheduled to be rebuilt, keeping your
F widget up-to-date.
T
ng: MediaQuery.of(context).size.width > 480
```





# O Método **Build** foi Chamado...

```
14 @override
15 Widget build(BuildContext context) {
16     return LayoutBuilder(
17         builder: (ctx, constraints) {
18             return Column(
19                 children: <Widget>[
20                     Container(
21                         height: constraints.maxHeight * 0.15,
22                         child: FittedBox(
23                             child: Text('${value.toStringAsFixed(2)}'),
24                         ), // FittedBox
25                     ), // Container
26                     SizedBox(height: constraints.maxHeight * 0.05),
27                     Container(
28                         height: constraints.maxHeight * 0.6,
29                         width: 10,
30                         child: Stack(
31                             alignment: Alignment.bottomCenter,
```

```
class Container extends StatelessWidget {
  /// Creates a widget that combines c
  ///
  /// The `height` and `width` values
  ///
```

**build**

