




Flexbox

Como fazer layouts no react native



Flexbox é uma maneira de organizar, alinhar e distribuir elementos dentro de um container mesmo sem saber o tamanho exato do container, por isso flex de flexível.



Render



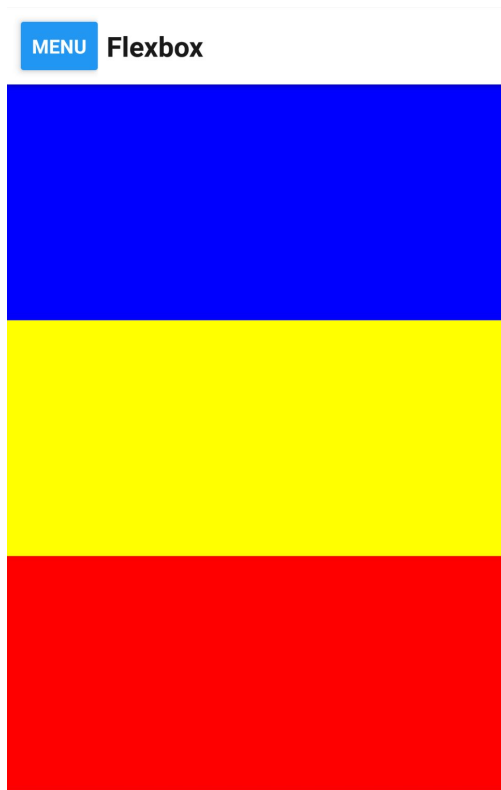
```
render() {  
  return (  
    <View style={styles.container}>  
      <View style={styles.blue} />  
      <View style={styles.yellow} />  
      <View style={styles.red} />  
    </View>  
  );  
}
```

Styles



```
const styles = StyleSheet.create({
  container: {
    // display: 'flex',
    flex: 1
  },
  blue: {
    backgroundColor: 'blue',
    flex: 1
  },
  yellow: {
    backgroundColor: 'yellow',
    flex: 1
  },
  red: {
    backgroundColor: 'red',
    flex: 1
  }
});
```

Resultado



flex



```
const styles = StyleSheet.create({
  container: {
    flex: 1
  },
  blue: {
    backgroundColor: 'blue',
    flex: 2
  },
  yellow: {
    backgroundColor: 'yellow',
    flex: 1
  },
  red: {
    backgroundColor: 'red',
    flex: 1
  }
});
```

Resultado

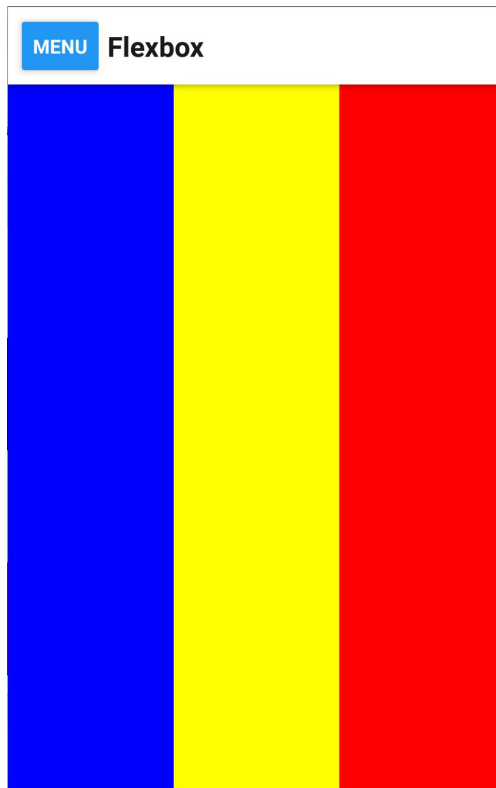


flexDirection



```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'row' // <-- flex direction
  },
  blue: {
    backgroundColor: 'blue',
    flex: 1
  },
  yellow: {
    backgroundColor: 'yellow',
    flex: 1
  },
  red: {
    backgroundColor: 'red',
    flex: 1
  }
});
```

Resultado



justifyContent



flex-end;

flex-start;

center; *

space-between; *

space-around; *

justifyContent



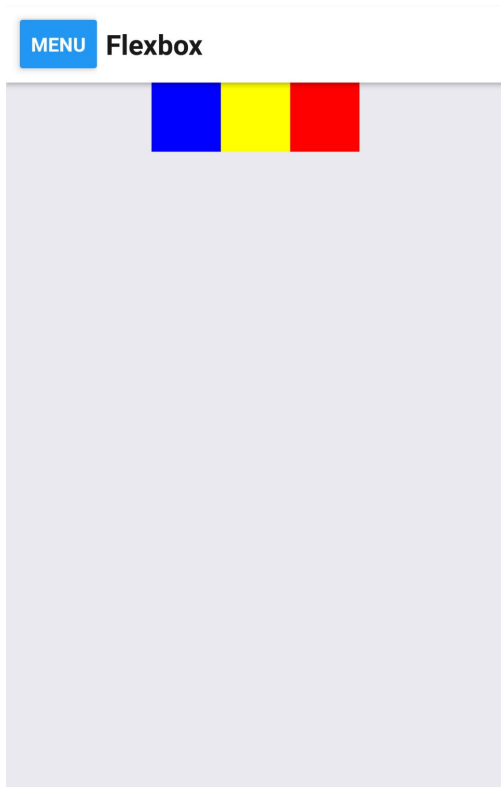
```
render() {  
  return (  
    <View style={styles.container}>  
      <View style={[styles.blue, styles.size]} />  
      <View style={[styles.yellow, styles.size]} />  
      <View style={[styles.red, styles.size]} />  
    </View>  
  );  
}
```

justifyContent

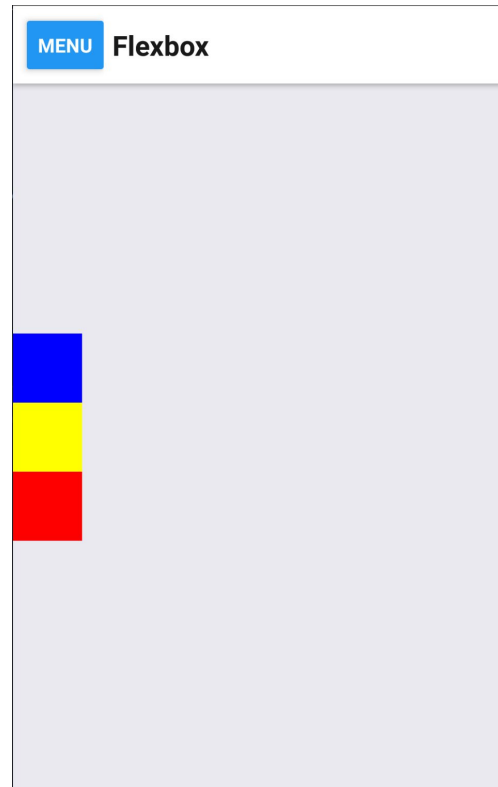


```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'row',
    justifyContent: 'center' // <-- justifyContent
  },
  size: {
    width: 50,
    height: 50
  },
  blue: {
    backgroundColor: 'blue'
  },
  yellow: {
    backgroundColor: 'yellow'
  },
  red: {
    backgroundColor: 'red'
  }
});
```

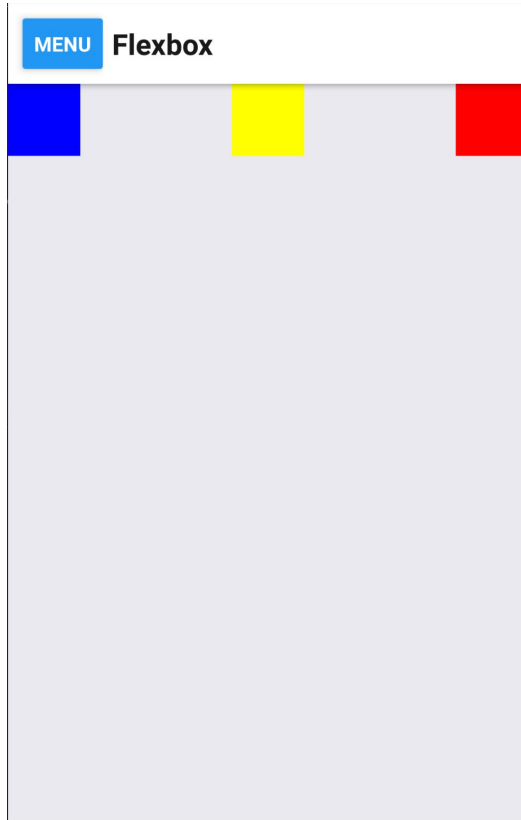
flexDirection: row;
justifyContent: center;



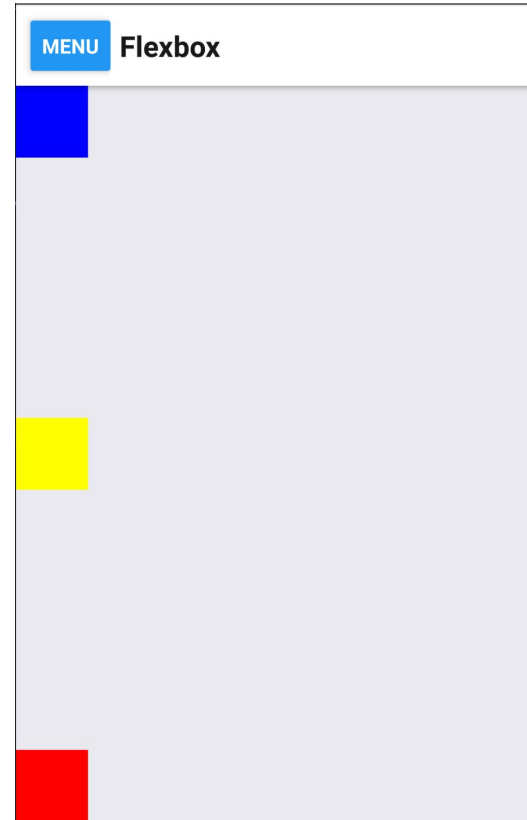
flexDirection: column;
justifyContent: center;



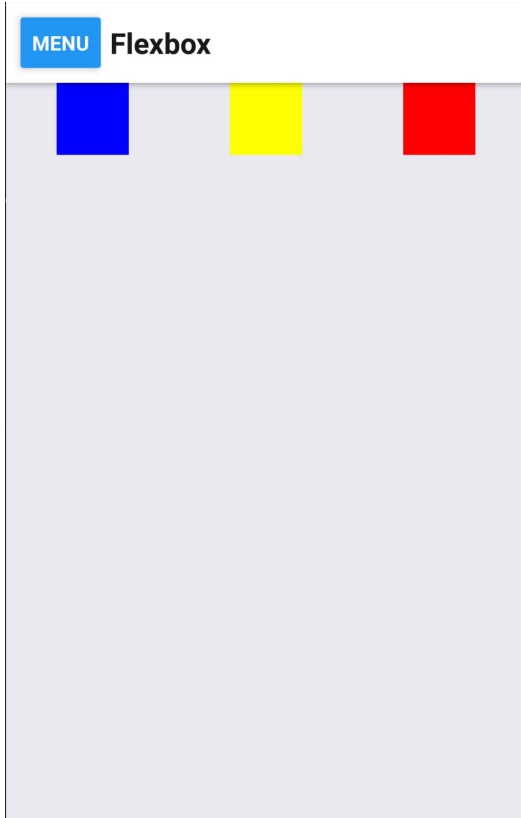
flexDirection: row;
justifyContent: space-between;



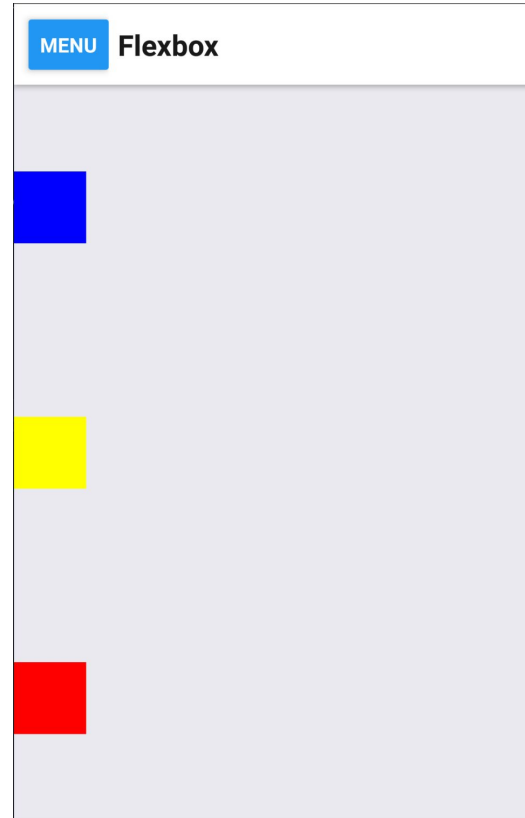
flexDirection: column;
justifyContent: space-between;



flexDirection: row;
justifyContent: space-around;



flexDirection: column;
justifyContent: space-around;



alignItems



flex-end;

flex-start;

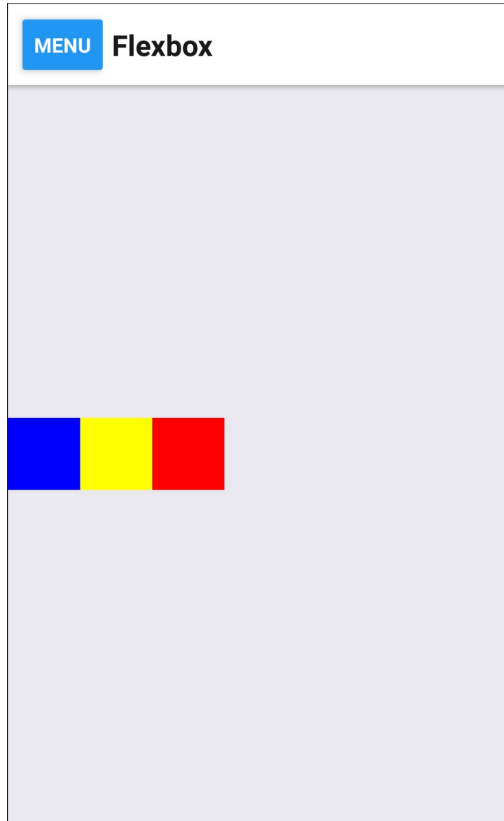
center; *

alignItems

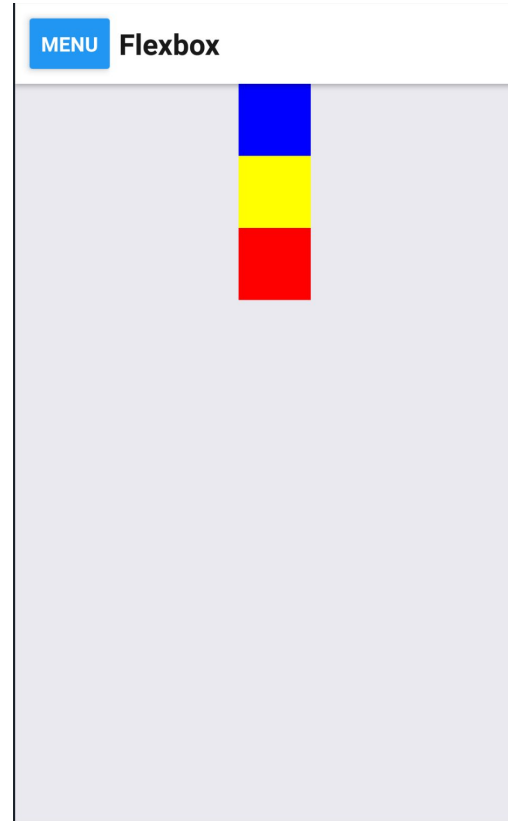


```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'row',
    alignItems: 'center', // <-- alignItems
  },
  size: {
    width: 50,
    height: 50
  },
  blue: {
    backgroundColor: 'blue'
  },
  yellow: {
    backgroundColor: 'yellow'
  },
  red: {
    backgroundColor: 'red'
  }
})
```

flexDirection: row;
alignItems: center;



flexDirection: column;
alignItems: center;



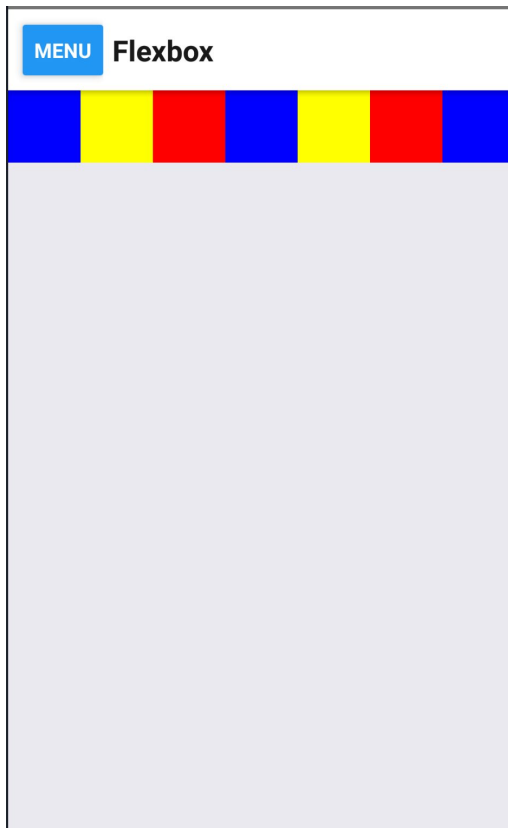
[illegible]

flexWrap

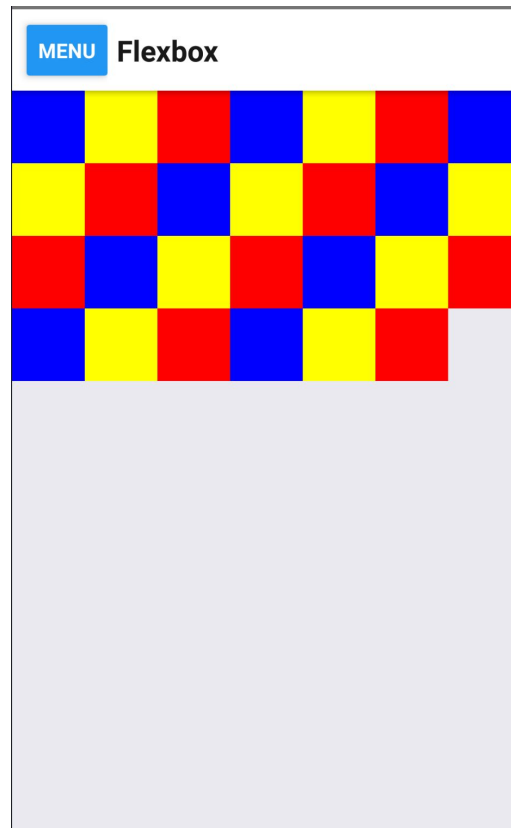


```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'row',
    flexWrap: 'wrap' // <-- flex Wrap
  },
  size: {
    width: 50,
    height: 50
  },
  blue: {
    backgroundColor: 'blue'
  },
  yellow: {
    backgroundColor: 'yellow'
  },
  red: {
    backgroundColor: 'red'
  }
});
```

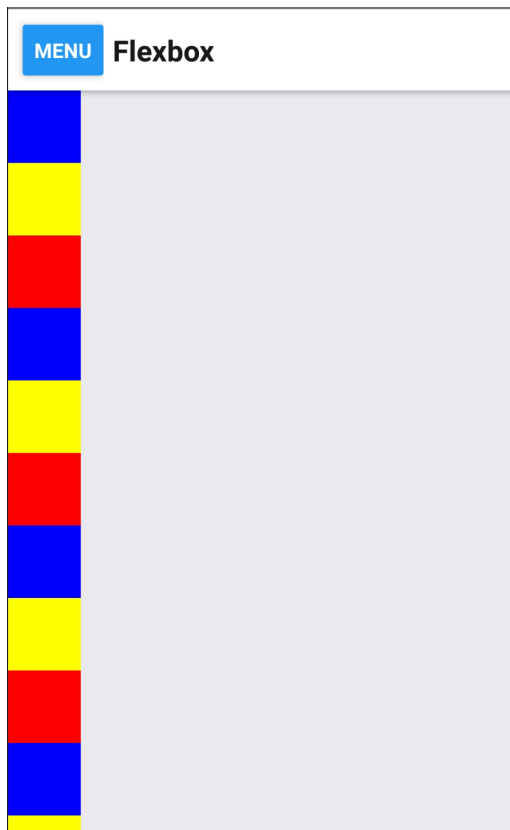
flexDirection: row;
Sem flex-wrap



flexDirection: row;
flexWrap: wrap;



flexDirection: column;
Sem flex-wrap



flexDirection: column;
flexWrap: wrap;

