



UICollectionView

Sergio Padrino Recio



UICollectionView

- Primera aparición en iOS 6 con clara orientación al iPad.
- Evolución de UITableView.

UICollectionView

Más opciones de personalización:

- Distribución en pantalla.
- Tamaño de celdas.
- Distancia entre celdas y secciones.
- Cabeceras y pies de sección.

UICollectionView

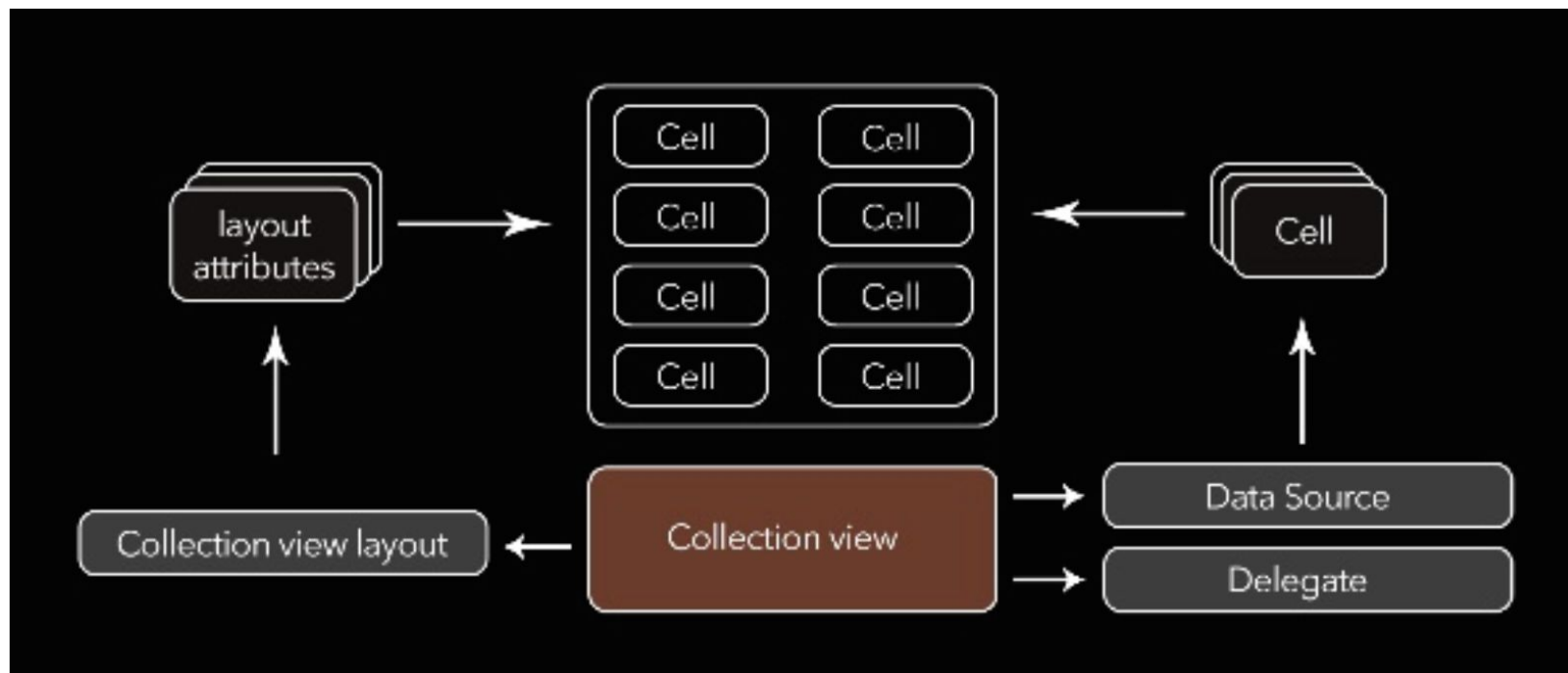
Diferente layout en las colecciones y las tablas:

- **Tablas:** un único layout con una única columna de celdas.
- **Colecciones:** configuración de un layout por defecto o creación desde cero.

UICollectionView vs UITableView

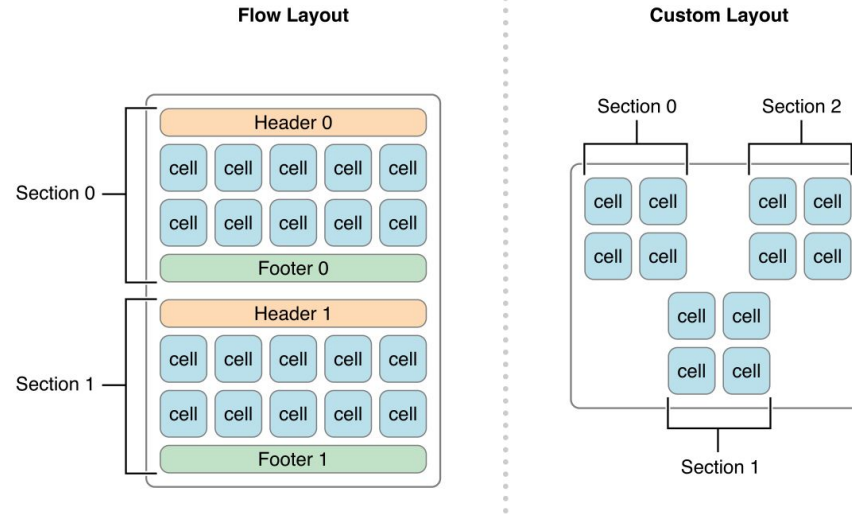
| | Colecciones | Tablas |
|------------------------|---|--|
| Clase base | UICollectionView | UITableView |
| Celda | UICollectionViewCell | UITableViewCell |
| Reciclar celdas | dequeueReusableCellWithReuseIdentifier(_:forIndexPath:) | dequeueReusableCellWithIdentifier(_:forIndexPath:) |
| Layout | UICollectionViewFlowLayout UICollectionViewLayout | Por defecto |
| Scroll | Vertical / Horizontal | Vertical |

UICollectionView: Integración



UICollectionView: Estructura

Depende del layout: Flow Layout o Custom



UICollectionView

Las colecciones utilizan tres delegates diferentes:

- **UICollectionViewDataSource:** permite a la colección obtener los datos que deben mostrar.
- **UICollectionViewDelegate:** permite a la colección notificar cuando ocurren eventos sobre la misma.
- **UICollectionViewDelegateFlowLayout:** permite a la colección configurar la apariencia de su contenido cuando se utiliza un Flow Layout.

UICollectionViewDataSource

Equivalentes a los métodos de UITableViewDataSource:

- `numberOfSectionsInCollectionView(_:)`
- `collectionView(_:numberOfItemsInSection:)`
- `collectionView(_:cellForItemAt:)`

UICollectionViewDelegate

Equivalente al método de UITableViewDelegate:

- collectionView(_:didSelectItemAt:)

UICollectionViewDelegateFlowLayout

- `scrollDirection`: vertical u horizontal
- `minimumLineSpacing`: interlineado
- `minimumInterItemSpacing`: espacio entre celdas
- `itemSize`: tamaño de celda
- `headerReferenceSize` / `footerReferenceSize`: tamaño de encabezado / pie

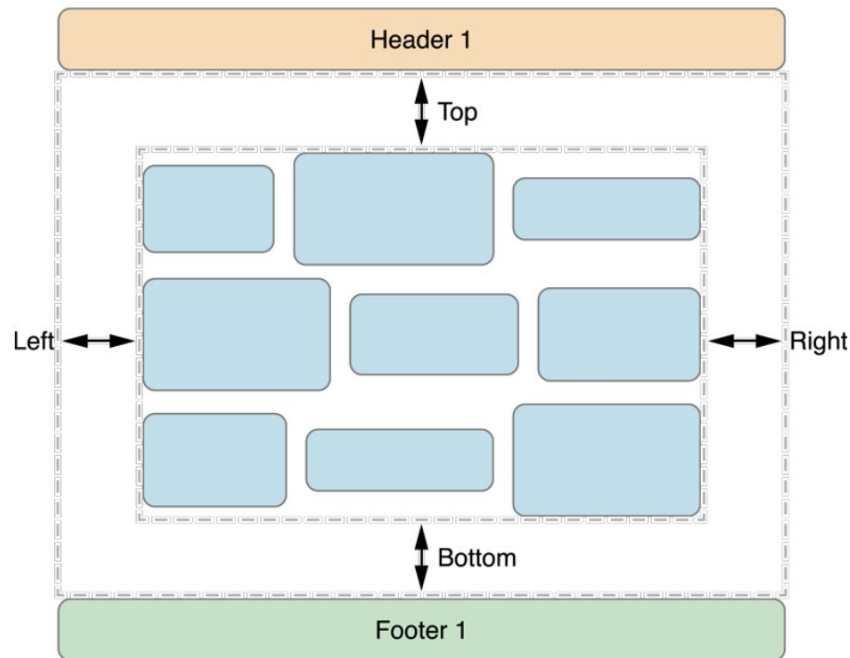
UICollectionViewDelegate

Tamaño de celda personalizado:

- `collectionView(_:layout:sizeForItemAt:)`

UICollectionView: sectionInset

```
inset = UIEdgeInsetsMake(top, left, bottom, right)
```



UICollectionViewCell

- Extender clase UICollectionViewCell para crear nuestra propia celda.
- Llamar a `registerClass(_:forCellWithReuseIdentifier:)` para poder utilizar la celda.
- Asignar un identificador si no utilizas Storyboard.

UICollectionView: Elegir layout

- Crear un layout:
`let myLayout = UICollectionViewFlowLayout()`
- Modificar propiedades del layout:
`myLayout.scrollDirection = .horizontal`
- Asignar el layout:
`collectionView.setCollectionViewLayout(myLayout,
animated: true)`

UICollectionView: Cabecera y pie

- Storyboard: activar *Accesories*
- Personalizar heredando de `UICollectionViewReusableView`

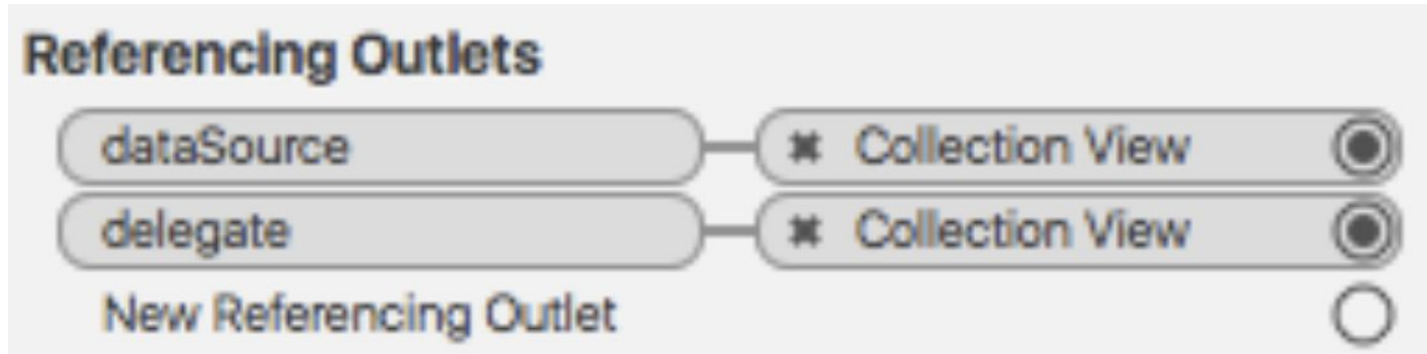
UICollectionView: Cabecera y pie

- `collectionView(_:viewForSupplementaryElementOfKind:atIndexPath:)`
- Posibles tipos:
 - `UICollectionViewElementKindSectionHeader`
 - `UICollectionViewElementKindSectionFooter`

UICollectionView paso a paso

UICollectionView: Paso a paso

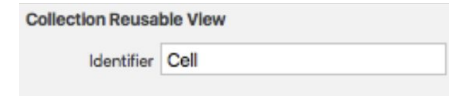
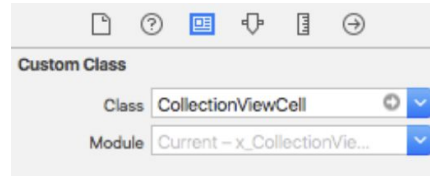
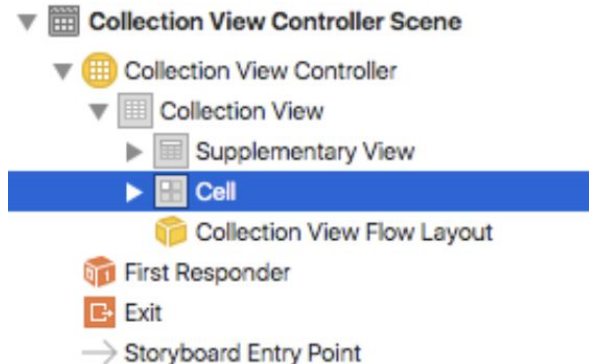
1. Añadir un UICollectionView en nuestro controller y asignarle su outlet.
2. Implementar la funcionalidad del dataSource y el delegado. Se asignan automáticamente al heredar de UICollectionViewController:



UICollectionView: Paso a paso

3. Añadir un prototipo de celda (con su propia clase custom), asignándole un identificador para su reutilización:

```
class UICollectionViewCell: UICollectionViewCell {
```

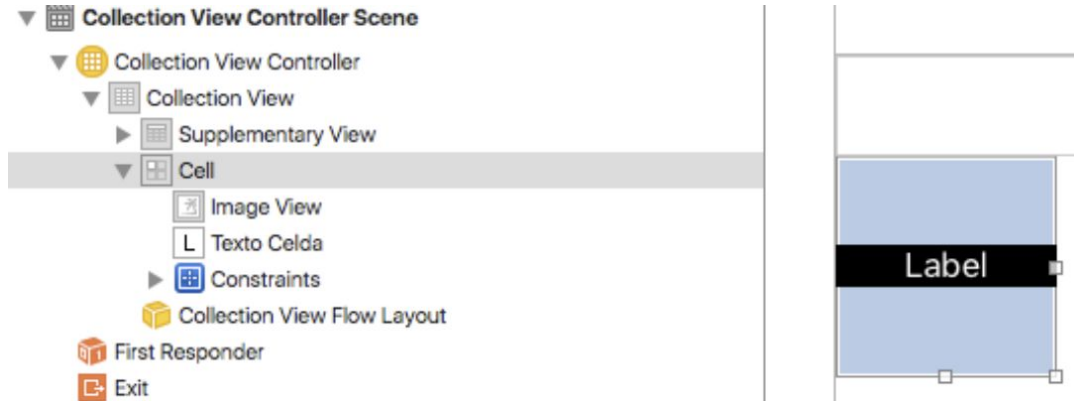


UICollectionView: Paso a paso

4. Diseñar la celda incluyendo sus elementos, layout...

```
@IBOutlet var imageView: UIImageView!
```

```
@IBOutlet var textoCelda: UILabel!
```



UICollectionView: Paso a paso

5. Implementar el modelo de datos:

```
let appNames = [  
    "Item 1",  
    "Item 2",  
    "Item N"  
]
```

UICollectionView: Paso a paso

5. Indicar número de secciones:

```
override func numberOfSectionsInCollectionView(collectionView: UICollectionView) -> Int
{
    return 1
}
```

UICollectionView: Paso a paso

5. Indicar número de elementos en la sección:

```
override func collectionView(collectionView: UICollectionView,  
                             numberOfItemsInSection section: Int) -> Int  
{  
    return appImages.count  
}
```


UICollectionView: Paso a paso

5. Configurar la celda:

```
override func collectionView(collectionView: UICollectionView,  
                             cellForItemAtIndexPath indexPath: NSIndexPath)  
    -> UICollectionViewCell  
{  
    ...  
}
```

UICollectionView: Paso a paso

5. Implementar funcionalidad adicional del UICollectionViewDelegate:

```
override func collectionView(collectionView: UICollectionView,  
                             didSelectItemAtIndexPath indexPath: NSIndexPath)  
{  
    ...  
}
```

UICollectionView: Paso a paso

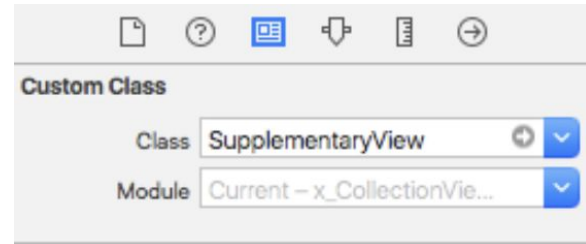
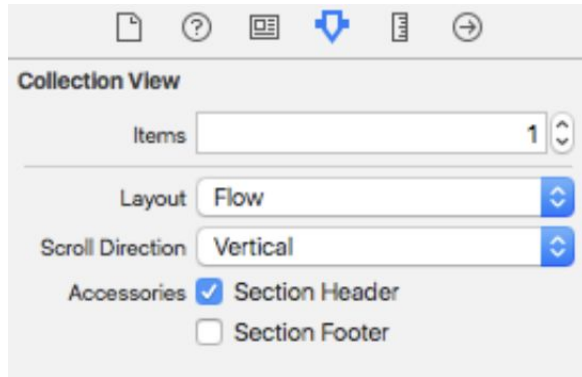
5. Implementar funcionalidad de UICollectionViewFlowLayoutDelegate:

```
func collectionView(_ collectionView: UICollectionView,  
                    layout collectionViewLayout: UICollectionViewLayout,  
                    sizeForItemAtIndexPath indexPath: IndexPath) -> CGSize  
{  
    ...  
}
```

UICollectionView: Paso a paso

Opcionalmente, puedes añadir cabecera/pie creando una clase que herede de `UICollectionViewReusableView`:

```
class SupplementaryView: UICollectionViewReusableView {
```



UICollectionView: Paso a paso

Y después crear una vista de este tipo en el método:

```
override func collectionView(_ collectionView: UICollectionView,  
                             viewForSupplementaryElementOfKind kind: String,  
                             atIndexPath indexPath: IndexPath) -> UICollectionViewReusableView  
{  
    ...  
}
```

UICollectionView: Paso a paso

Para modificar el layout, debes crear un layout, cambiar sus propiedades y asignarlo a tu Collection View:

```
let myLayout = UICollectionViewFlowLayout()
```

```
myLayout.scrollDirection = .horizontal
```

```
myLayout.sectionInset = UIEdgeInsetsMake(100, 5, 10, 10)
```

```
myLayout.minimumInteritemSpacing = 50
```

```
collectionView.setCollectionViewLayout(myLayout, animated: true)
```