

Tema 7

Creación de interfaces gráficas II

Controlador de controladores



Controlador de controladores

- Por norma general los controladores se encargan de gestionar el modelo y su vista o vistas asociadas, pero existen una serie de controladores especiales cuya función consiste en gestionar un conjunto de View Controllers.
- Dentro de estos controladores especiales tenemos tres tipos:
 - UINavigationController
 - UITabBarController
 - UISplitViewController

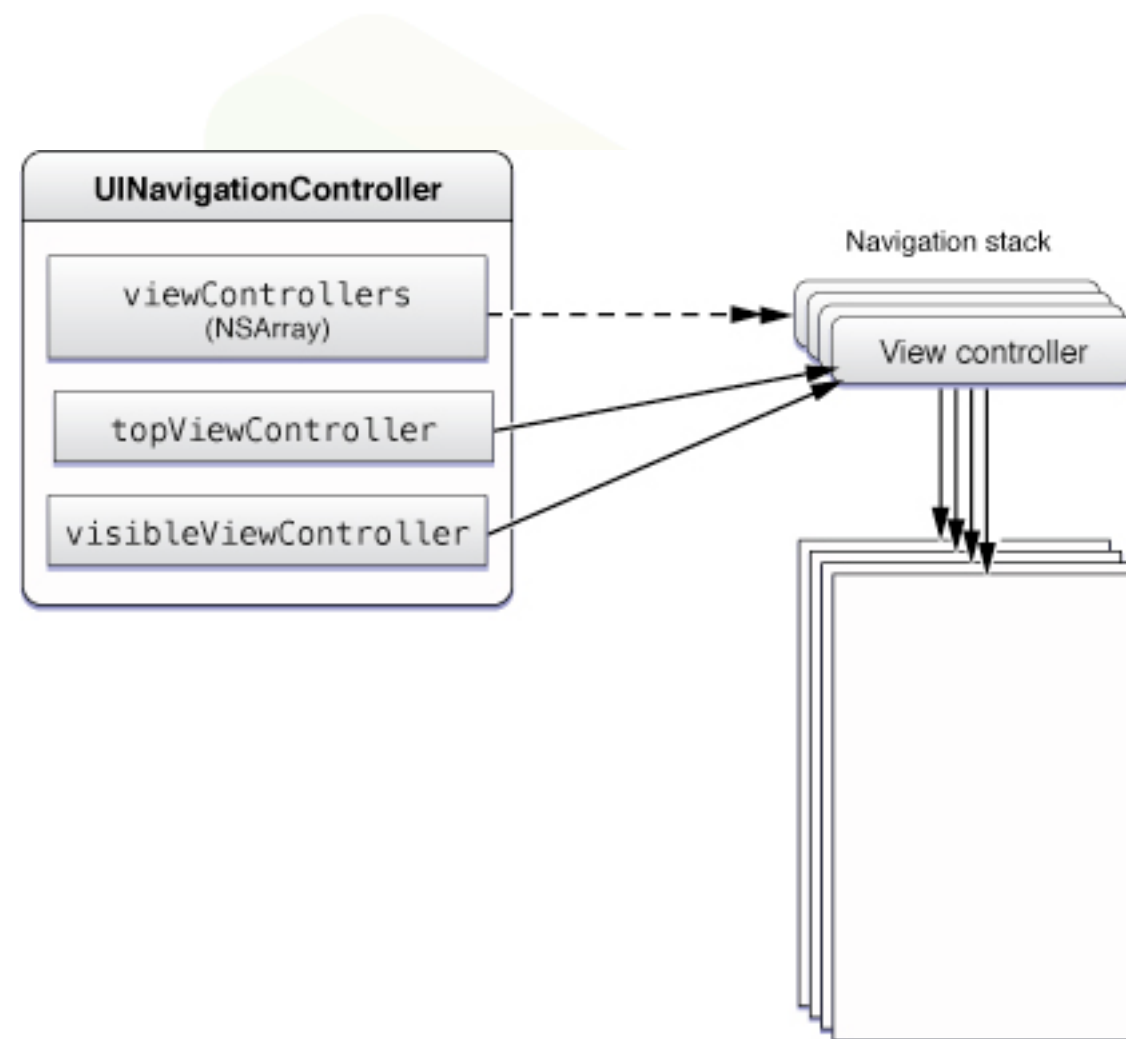
UINavigationController



UINavigationController

- **UINavigationController** es un controlador encargado de gestionar una serie de controladores entre los cuales se establece una relación de navegación por lo que de un controlador se pasa a otro y desde el último es posible navegar al anterior.
- Dispone de un NSArray de controladores donde se almacenan los controladores por los que se ha pasado, desde el inicial, hasta el que se muestra.
- El uso del controlador es igual al de una pila. En este caso, se apilan y desapilan controladores, que se añaden y eliminan mediante **push** y **pop**.
- El controlador inicial, conocido como **root view controller**, no puede desapilarse o eliminarse.

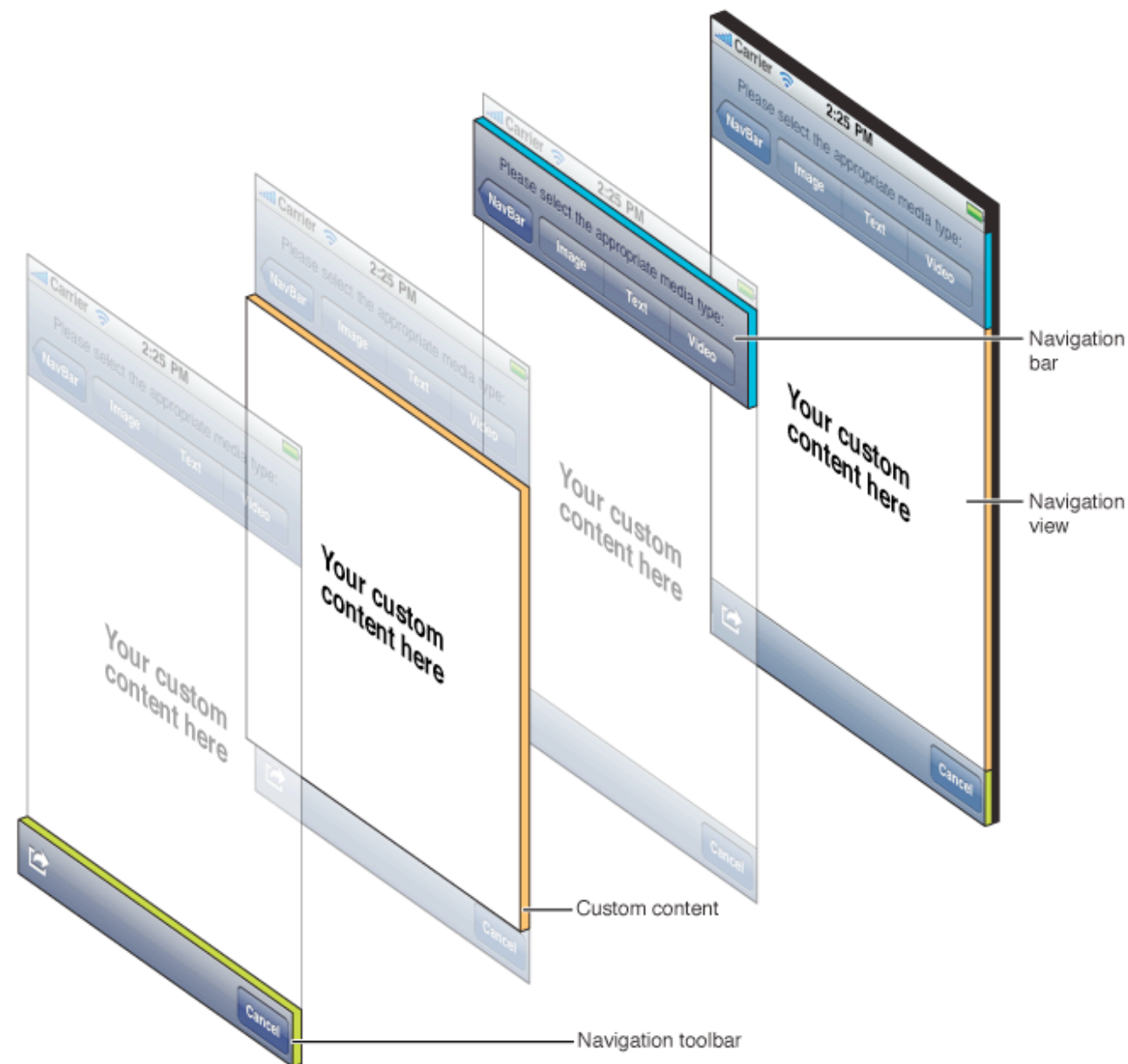
UINavigationController



UINavigationController

- El controlador dispone de una estructura en capas.
- La primera capa, situada por debajo del resto, es la capa referente a la vista del UINavigationController.
- Por encima de esta se sitúa la barra de navegación, conocida como **navigation bar**.
- La tercera capa corresponde al contenido propio, es decir, a las vistas utilizadas por el usuario, que se van apilando en la pila del controlador.
- Por último, y de forma opcional se puede incluir una barra inferior, conocida como **navigation toolbar**.

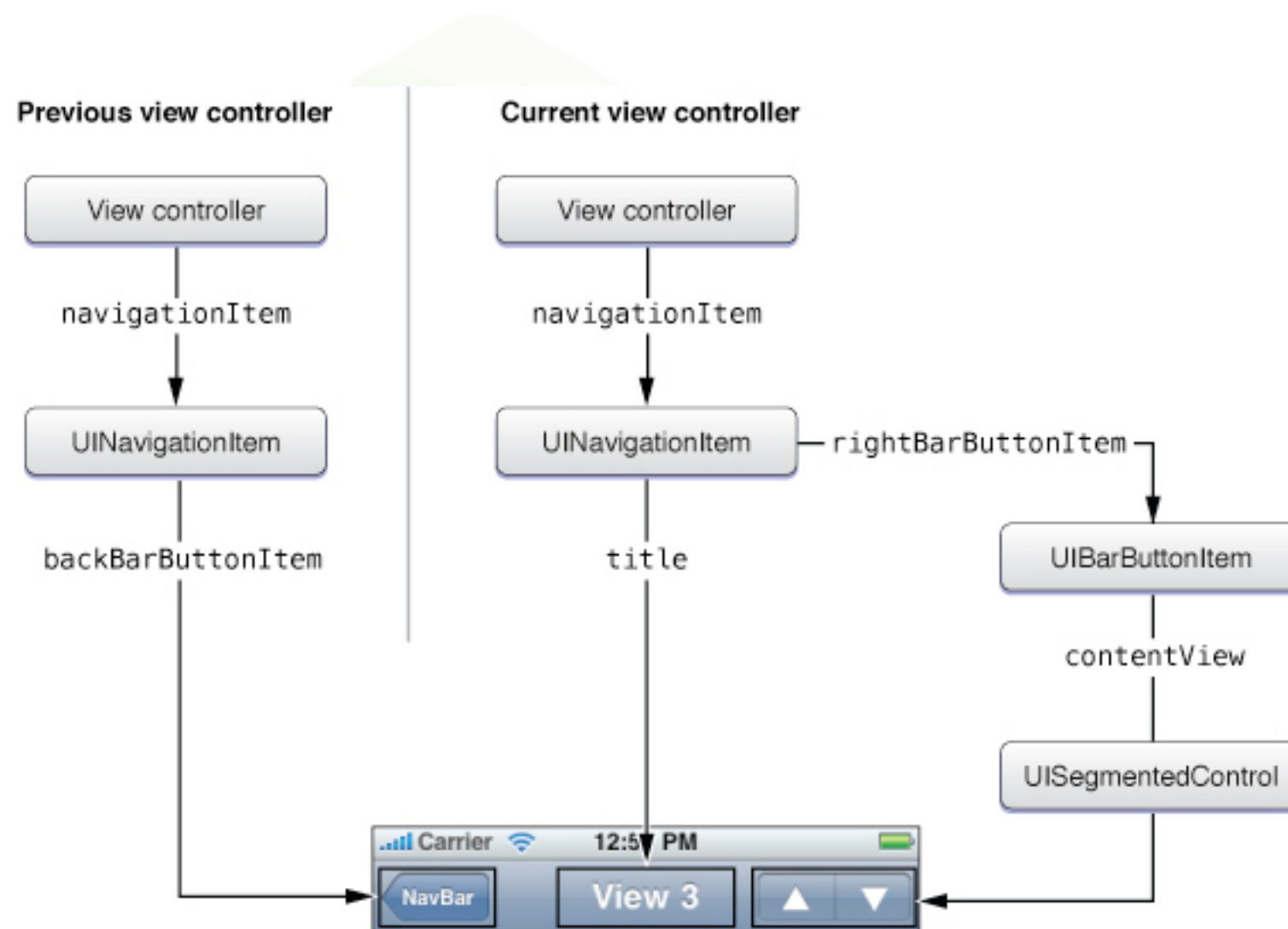
UINavigationController



UINavigationController

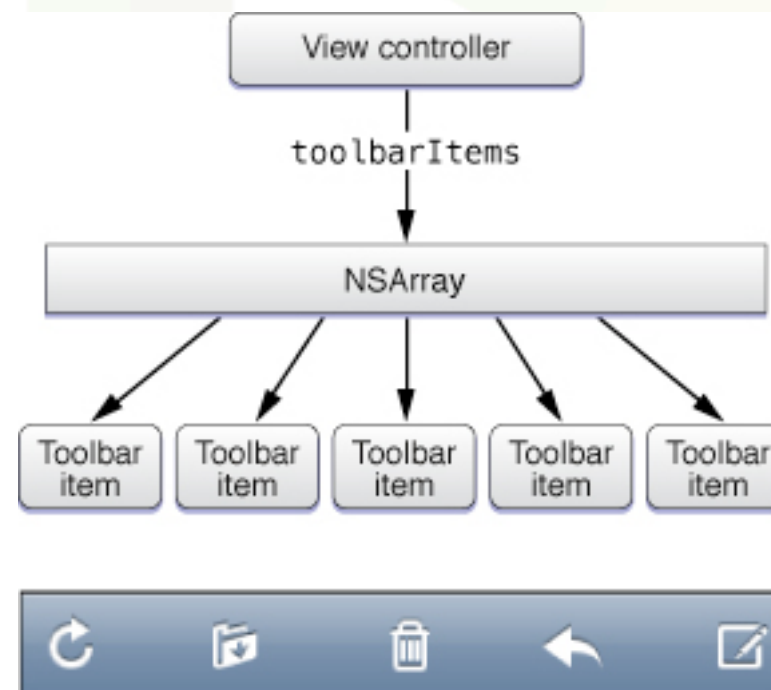
- La **navigation bar** se compone de elementos **UINavigationControllerItem**. Existen tres opciones a la hora de añadir elementos a la barra de navegación.
- La **posición izquierda** de la barra se utiliza para el botón volver por normal general, aunque puede utilizarse para un botón propio. Se accede mediante **backBarButtonItem** o **leftBarButtonItem**.
- La **posición central** corresponde con el título. Se accede mediante **titleView** y se suele utilizar para proporcionar información al usuario de dónde se encuentra.
- La posición derecha se utiliza para añadir los botones necesarios según se desee. Se accede mediante **rightBarButtonItem**.

UINavigationController



UINavigationController

- La barra inferior llamada **navigation toolbar** o toolbar tiene una estructura de elementos llamados **toolbar item**, almacenados en un NSArray.



UINavigationController

- A la hora de crear un UINavigationController podemos crearlo como controlador inicial mediante la opción **Master-Detail Application** al crear un nuevo proyecto. Esta opción en iPad creará una vista dividida conocida como **UISplitViewController**.
- Otra opción es utilizarlo como pestaña en un **UITabBarController**.
- Este tipo de controlador también puede presentarse de forma **modal** desde otro controlador.

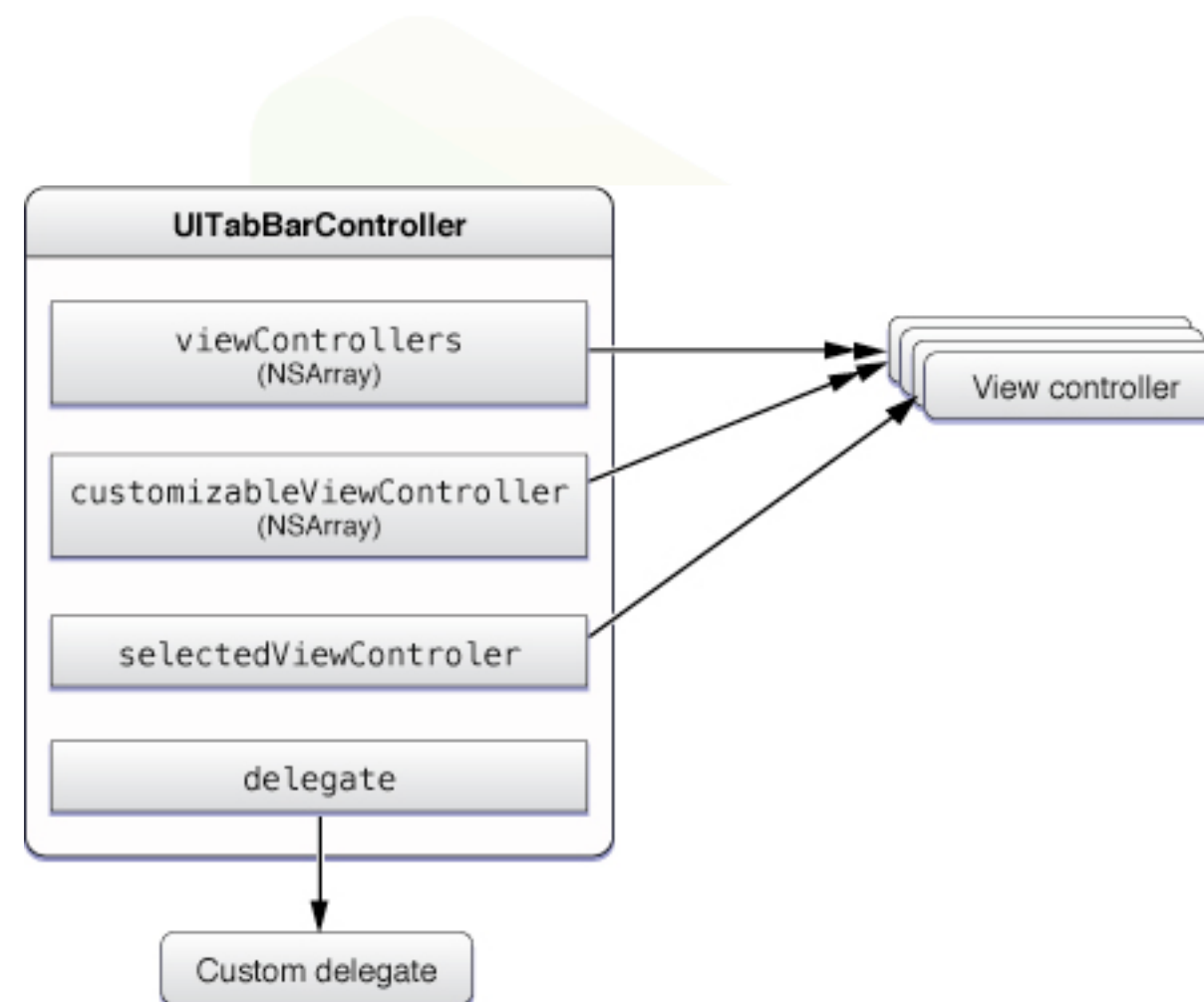
UITabBarController



UITabBarController

- **UITabBarController** gestiona una serie de controladores mediante el uso de pestañas o tabs. A diferencia del UINavigationController, no existe relación entre los controladores.
- También dispone de un NSArray de controladores donde se almacena cada controlador correspondiente a cada una de las pestañas.

UITabBarController



UITabBarController

- La estructura de este tipo de controlador es similar a la del UINavigationController.
- La primera capa corresponde también a la vista del controlador, en este caso del tab bar controller.
- La segunda capa corresponde al contenido propio, donde se encuentra cada una de las vistas de cada una de las pestañas.
- Por último, por encima del resto de capas se encuentra la barra inferior llamada **tab bar**, que contiene los botones para acceder a las pestañas.

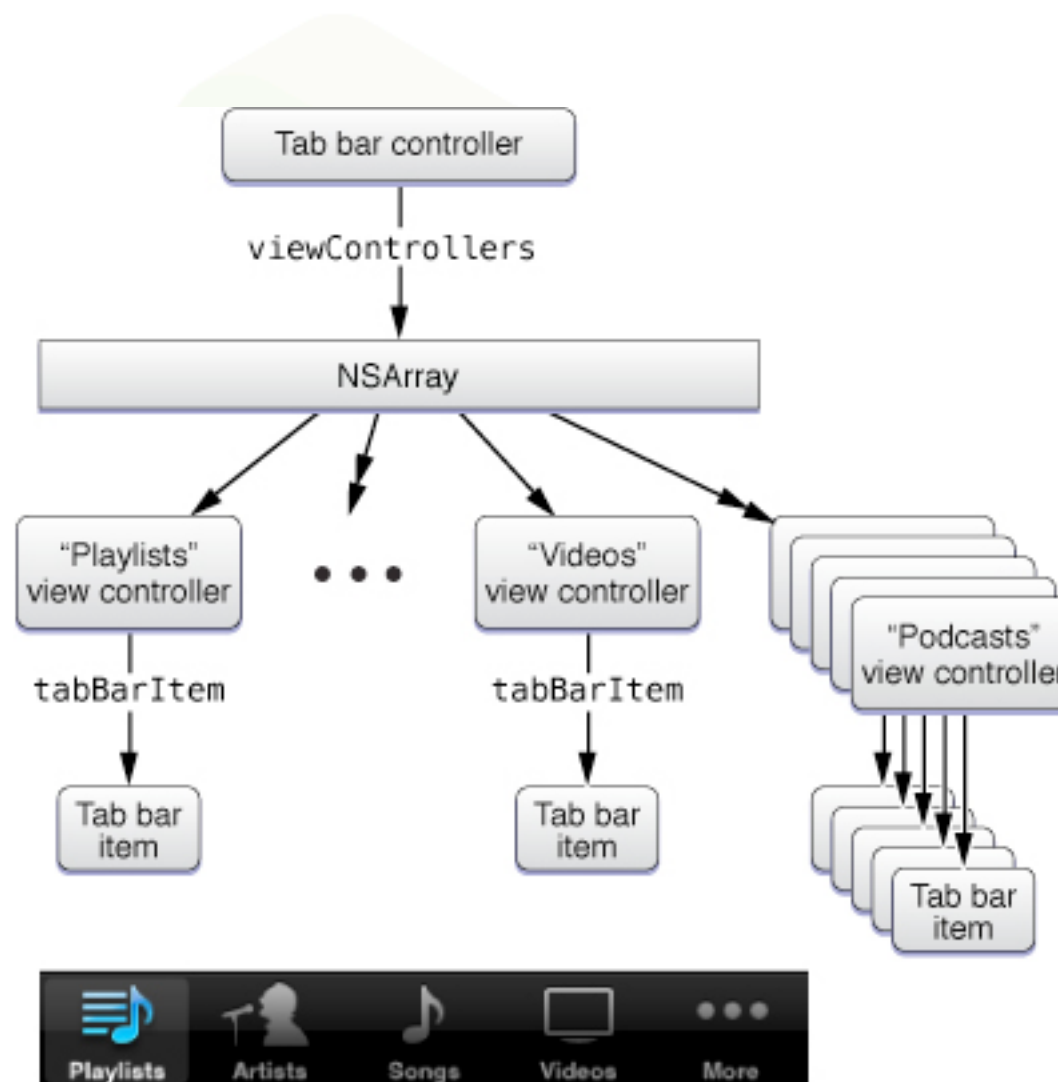
UITabBarController



UITabBarController

- La **tab bar** se compone de elementos **UITabBarItem**.
- Estos elementos pueden tener una imagen, un nombre y además es posible indicar un valor para que lo muestren como alerta sobre la pestaña correspondiente.

UITabBarController



UINavigationController

UITabBarController

UINavigationController - UITabBarController

- Existe la posibilidad de combinar el uso de un UINavigationController y un UITabBarController.
- Para conseguir esta característica el controlador base debe ser el UITabBarController y en una o varias de sus pestañas se añadiría el UINavigationController que se quisiese.

UICollectionView

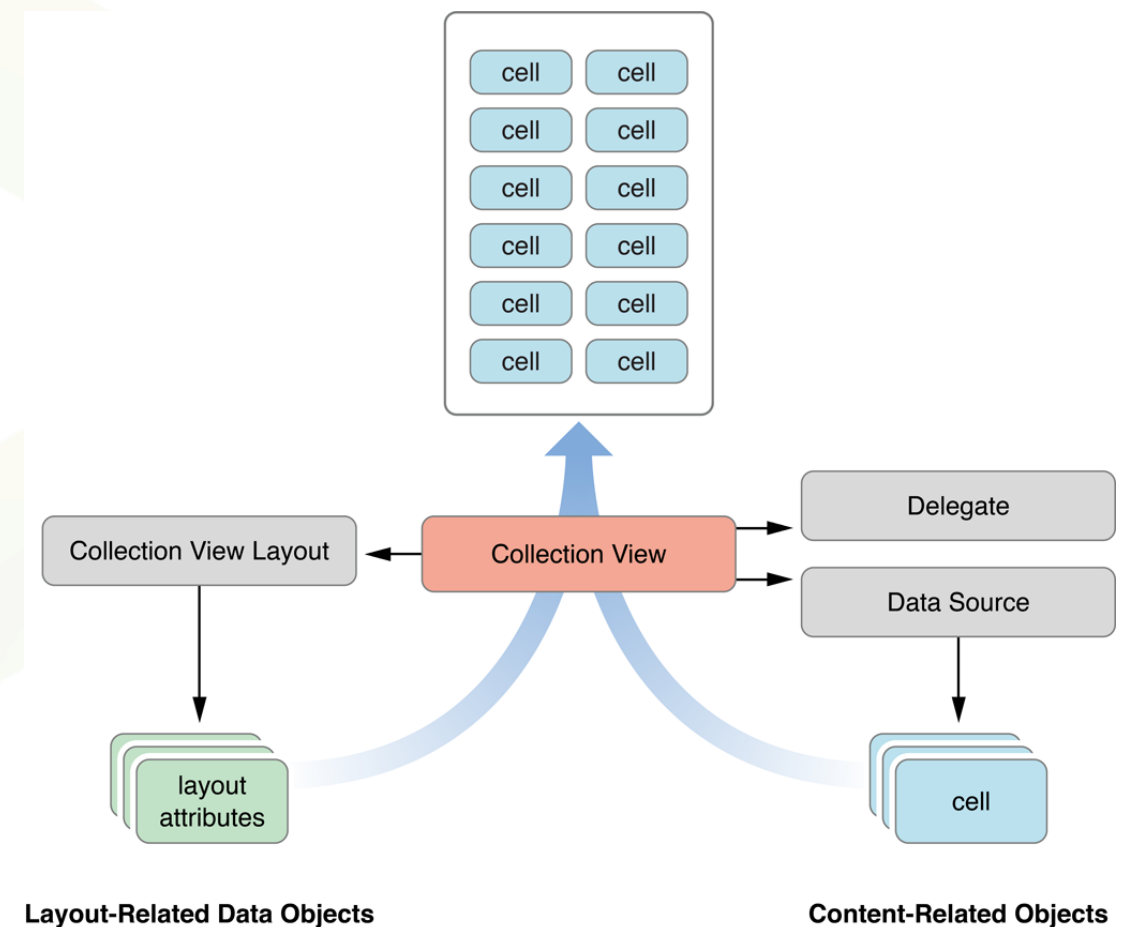


UICollectionView

- **UICollectionView** es un view controller que permite organizar vistas en forma de cuadrícula.
- El funcionamiento se basa en tres delegados.
- El primero, **UICollectionViewDelegate**, es el encargado de controlar la selección y deselección de elementos.
- **UICollectionViewDataSource** es el encargado de proporcionar el contenido al controlador. Indica el número de elementos y secciones que se mostrarán entre otras cosas.
- Por último, **UICollectionViewDelegateFlowLayout** gestiona la distribución de los elementos en la cuadrícula. Es útil cuando queremos emplear una distribución distinta a las proporcionadas por defecto.

UICollectionView

- Únicamente es obligatorio el uso del data source, puesto que necesitamos indicar los elementos que se mostrarán y qué apariencia tendrá.
- Mediante estos tres delegados se crea y gestiona la colección de vistas.



UICollectionView

- En el uso de controladores como UICollectionView es importante hacer uso de la reutilización de vistas ya que es posible que se muestren multitud de elementos y de lo contrario dispondríamos de todos los elementos que se podrían mostrar cargados en memoria.
- Para conseguir la reutilización, el controlador elimina las vistas que no se están mostrando y las almacena en un set para poder ser eliminadas.
- En el momento en el que se quiere mostrar una nueva vista, se obtiene una vista de las eliminadas y se reutiliza, evitando tener que crear una nueva instancia.

UICollectionView

- Este proceso de reutilización de vistas se realiza en el método delegado **collectionView:cellForItemAtIndexPath:.** En este método se genera la vista correspondiente a una posición en la cuadrícula, por lo que espera un elemento como valor de retorno.
- Dentro del método debemos obtener una vista para poder reutilizarla mediante el método **dequeueReusableCellWithIdentifier:forIndexPath:.**
- Una vez obtenida la celda se modifica para actualizar sus datos y que correspondan a la posición indicada.

Auto Layout

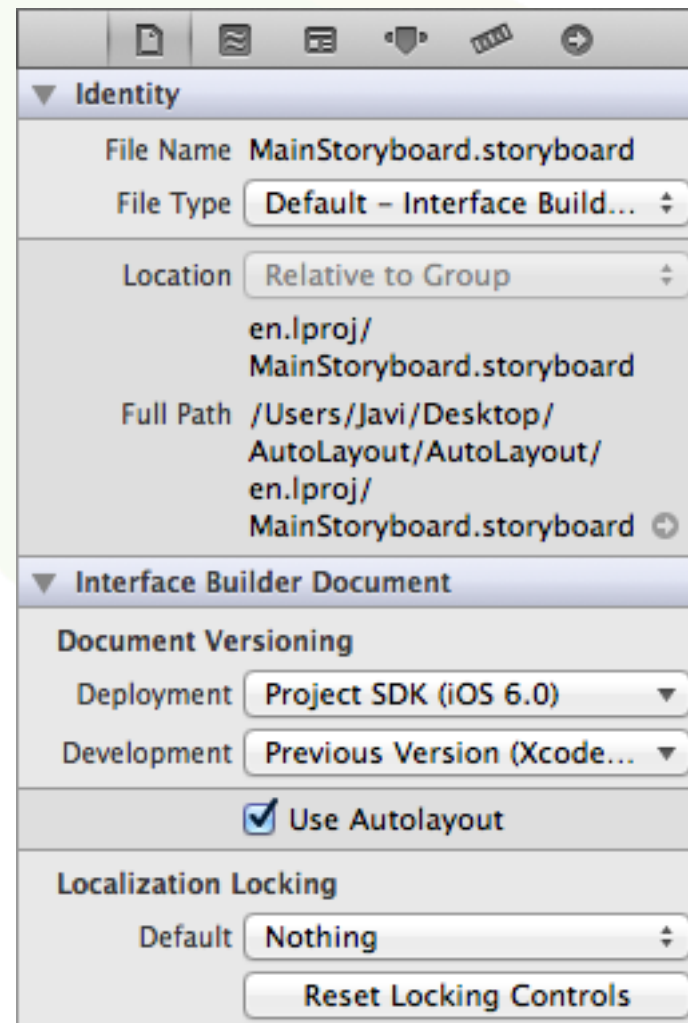


Auto Layout

- **Auto Layout** es una característica que permite añadir elementos a las interfaces gráficas que cumplan una serie de restricciones.
- Ejemplos de estas restricciones son por ejemplo, mantener cierta distancia con la parte superior o inferior de la vista, mantener una separación entre diversos elementos de la interfaz, etc.
- Es útil a la hora de desarrollar interfaces en horizontal y vertical o de adaptar aplicaciones a dispositivos con una pantalla de tamaño distinto como el iPhone 5.

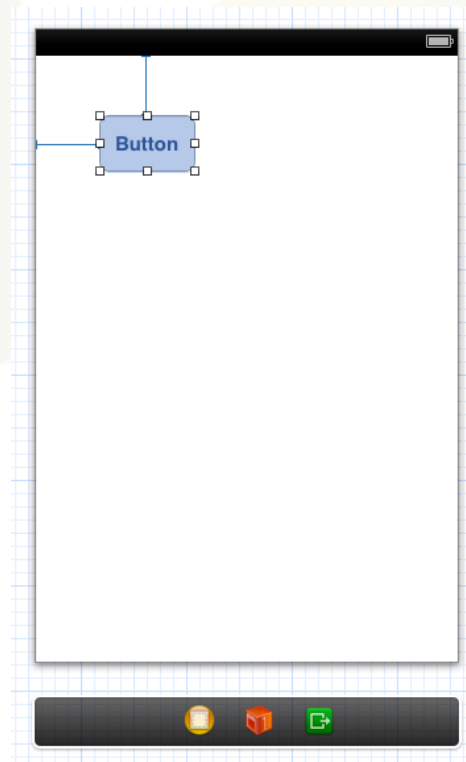
Auto Layout

- Es una opción disponible en archivos XIB y storyboards.



Auto Layout

- Al añadir un elemento a la interfaz mientras auto layout está activado se incorporan una serie de restricciones al elemento según la posición donde se añada dicho objeto.
- Las restricciones se identifican por unas líneas al seleccionar un elemento de la interfaz.



Auto Layout

- Las restricciones pueden ser modificadas para cambiar el valor correspondiente a la constante que se debe cumplir, así como si debe ser igual al valor, mayor o menor, según nos interese.
- También podemos seleccionar prioridad para las restricciones, de forma que nos aseguramos que una restricción tendrá mayor prioridad que otra y por tanto deberá cumplirse de forma más prioritaria frente a otras.