



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS

Split View Controller

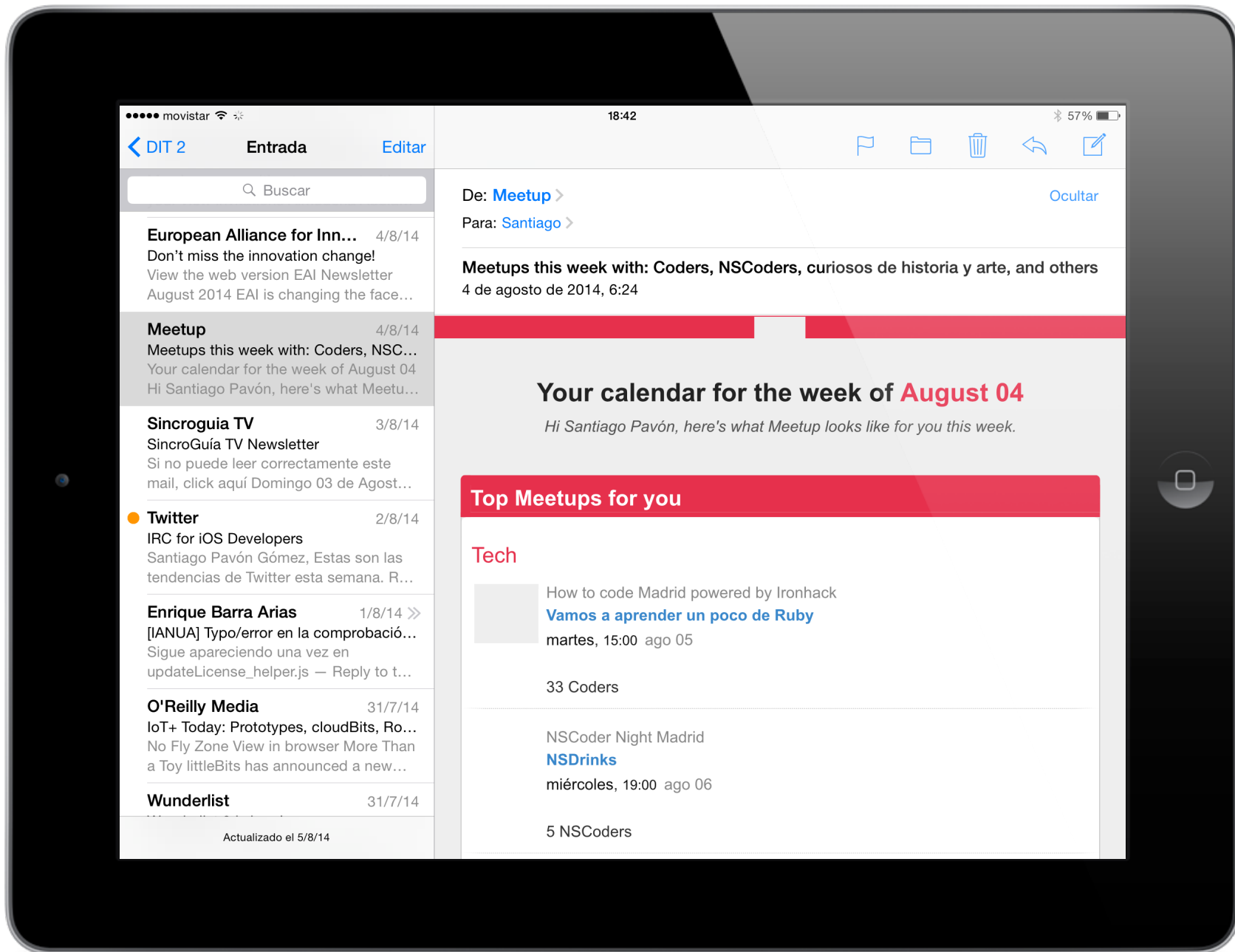
IWEB,LSWC 2014-2015

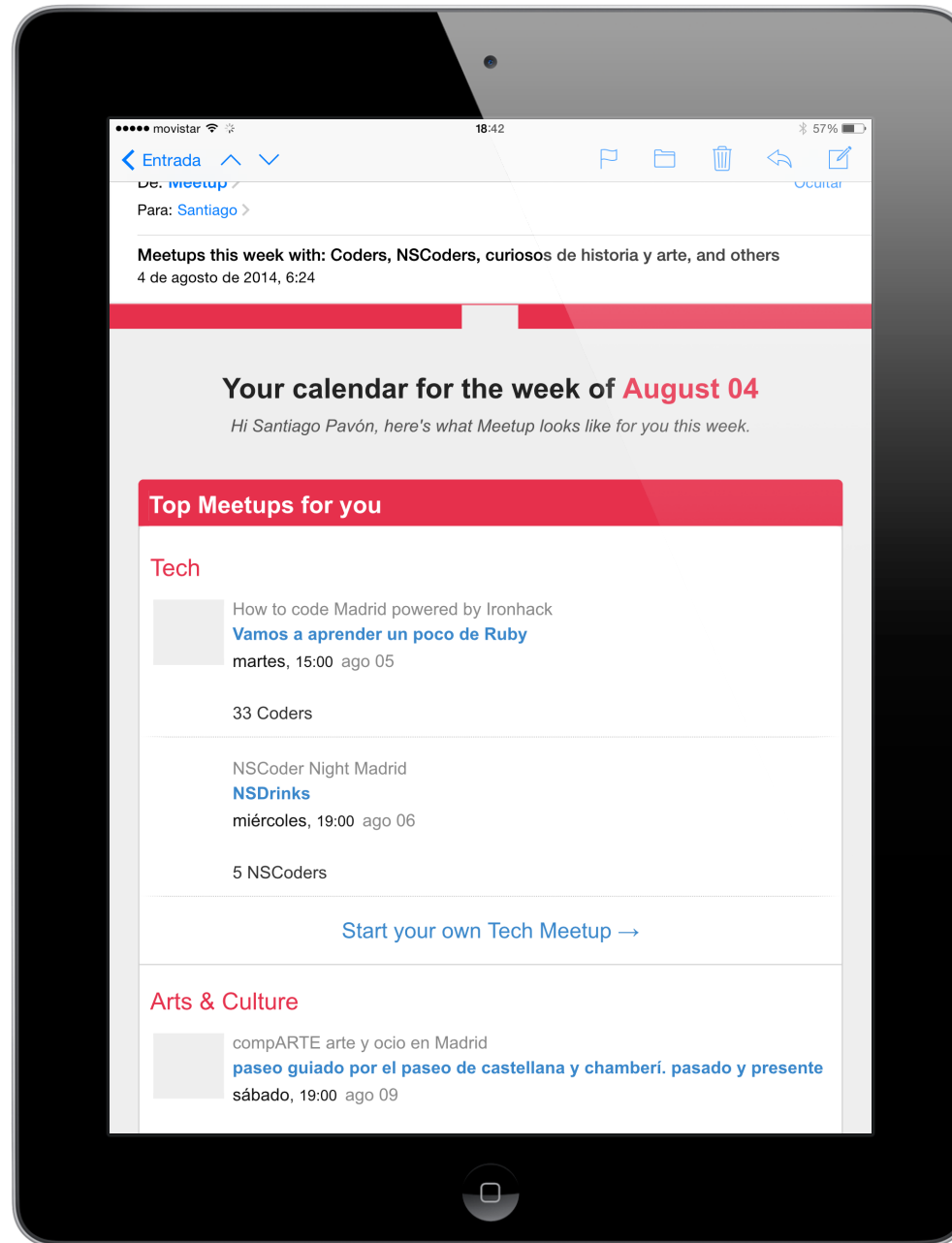
Santiago Pavón

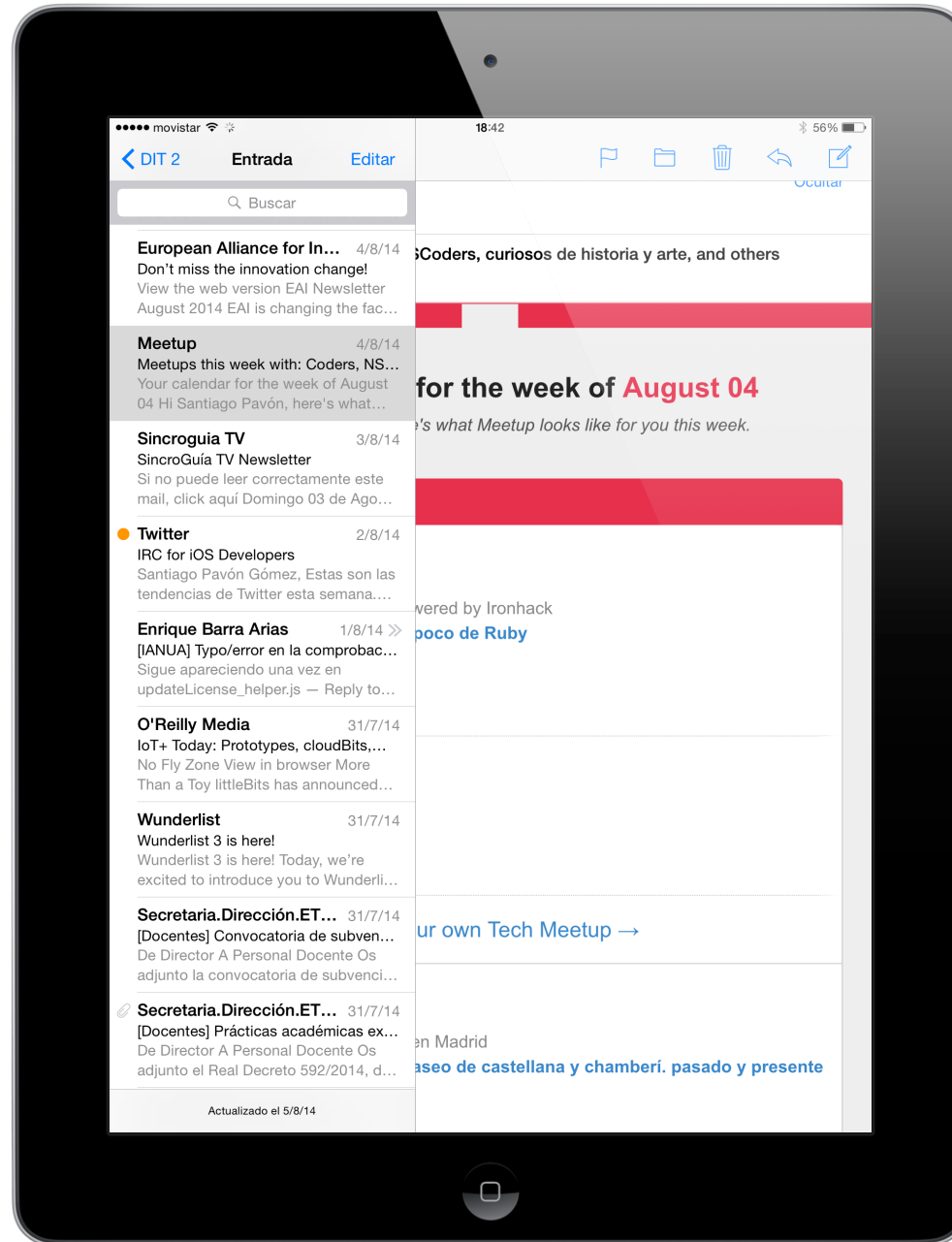
ver: 2014.11.06

UISplitViewController

- UISplitViewController permite ver dos UIViewControllers simultáneamente: uno llamado **master** y otro llamado **detail**.
- En iOS 7 y anteriores solo se soportaba en terminales iPad.
- Con iOS 8 ya se soporta en todo tipo de terminales, iPhones y iPads.
 - Sin embargo, su presentación se adapta dependiendo del Trait Collection del terminal.
 - Así, por defecto, en un iPhone se muestra como si estuviera implementado con Navigation Controllers.



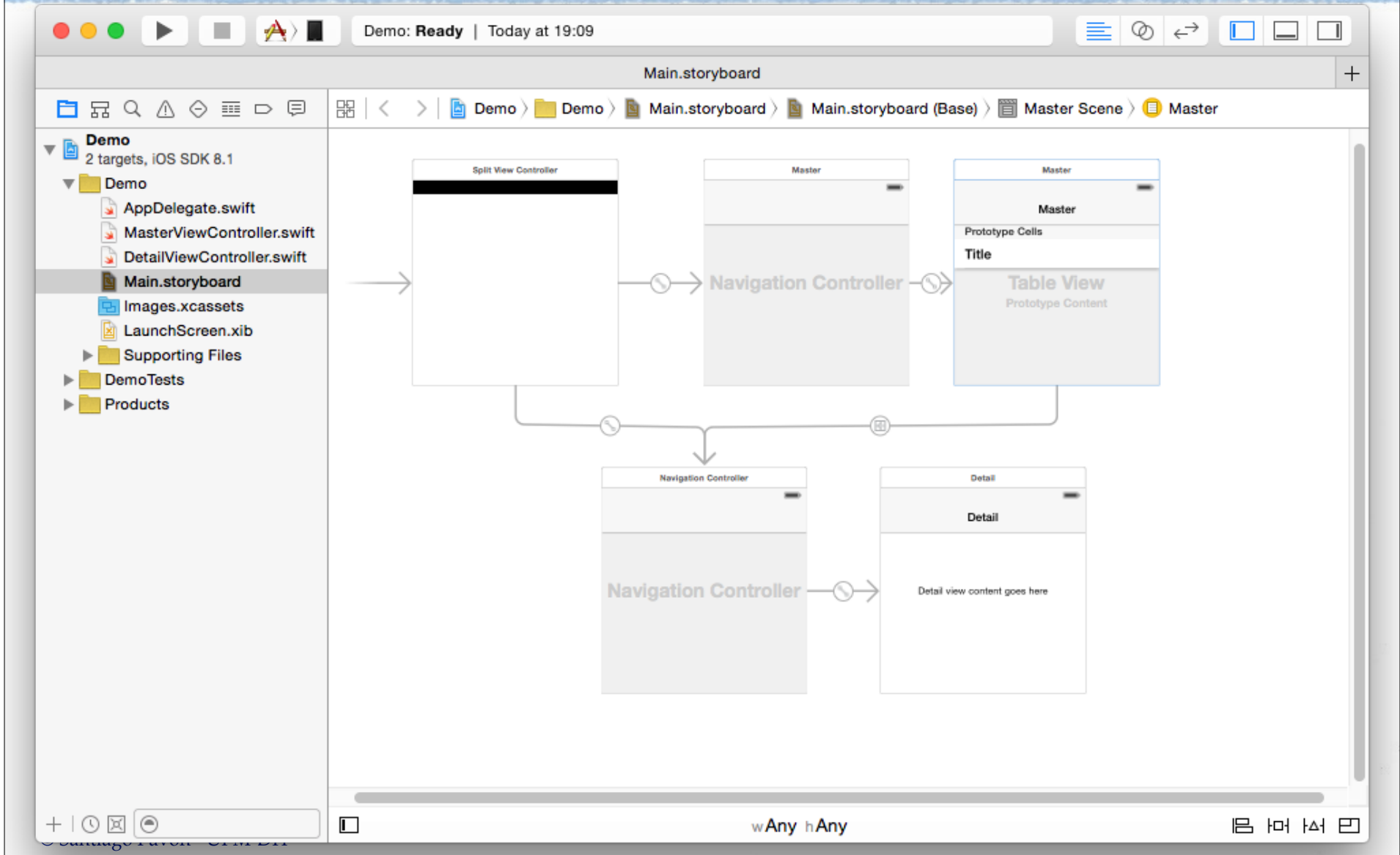




Crear Split View Controller

- Un objeto UISplitViewController puede crearse:
 - Programáticamente:
 - Crear objeto UISplitViewController, asignar a la propiedad **viewController** los hijos VC a mostrar como master y detail, asignar el delegado y sobrescribir los métodos necesarios, etc.
 - Usando Xcode para crear un nuevo proyecto basado en la plantilla **Master-Detail Application**:
 - Crea un proyecto que usa un Split View Controller con una configuración inicial suficiente.
 - Se configura en el delegado de la aplicación.
 - Usando Interface Builder para arrastrar desde la librería de objetos hasta el storyboard un objeto Split View Controller.
 - Es necesario enganchar el delegado del SVC y configurar numerosos detalles.

Plantilla: Master-Detail App



- El Split View Controller tiene dos segues de tipo **Master View Controller** y **Detail View Controller** hacia sus dos hijos.
- Los VCs master y detail se presentan bajo el control de dos Navigation Controller.
- El delegado del Split View Controller es AppDelegate, es decir, el delegado de la aplicación.
- Desde las celdas de la tabla del VC master se ha creado un segue de tipo **Show Detail**.
 - Sustituye el objeto DetailVC mostrado por el Navigation Controller por una nueva instancia.
- La plantilla nos proporciona los ficheros MasterVC.swift y DetailVC.swift para las clases de los VC creados por la plantilla.

Uso de Segues

- Cuando se usa un Split View Controller pueden usarse distintos tipos de segue:
 - Un segue de tipo **Show**
 - Presentar un nuevo VC controlado por mi Navigation Controller, tanto si estoy en la zona master o en la detail.
 - Un segue de tipo **Show Detail**
 - Presentar un nuevo VC como VC Detail del Split View Controller situado más arriba en la jerarquía de View Controllers.
 - Un segue de tipo **Present Modally**
 - Presentar un nuevo VC de forma modal.
 - Un segue de tipo **Present as Popover**
 - Presentar un nuevo VC en un popover.
 - Un segue de tipo **Unwind**
 - Retroceder al VC del método apuntado por el segue.

Sustituir los VC Hijos

- Dos métodos muy útiles introducidos en iOS 8 son:

showDetailViewController:sender:

- Si un VC usa este método, se busca el SVC que lo contiene, y se sustituye el VC Detail actual por el pasado como parámetro.

showViewController:sender:

- Si un VC usa este método, se busca el SVC o el Navigation Controller que lo contiene,
 - si es un SVC, se sustituye el VC Master actual por el pasado como parámetro.
 - si es un Navigation Controller, se añade el VC pasado como parámetro a la pila de navegación.

- Este es el comportamiento por defecto de los VC y controladores existentes, que usan el método **targetViewControllerForAction:sender:** para buscar quien responde a los métodos anteriores.
 - Este método puede sobrescribirse en nuestros controladores para implementar el comportamiento que se desee.

Display Mode

- Usar la propiedad **preferredDisplayMode** para indicar el modo en el que se desea mostrar un Split View Controller **expandido**:
 - **.AllVisible**
 - iPad en horizontal.
 - **.PrimaryHidden**
 - iPad en vertical.
 - **.PrimaryOverlay**
 - iPhone 6 Plus en horizontal.

(**Expandido** significa que el SVC puede mostrar simultáneamente su VC master y su VC detail.)
- El modo actual con el que se está presentando un SVC lo indica la propiedad **displayMode**.
- Los SVC tienen un método llamado **displayModeButtonItem()** que devuelve un Bar Button Item para cambiar su display mode.
 - Este botón solo funciona cuando el SVC está expandido.
 - La plantilla Master-Detail App coloca este botón en la barra de navegación del VC detail.
 - Este botón solo es visible en:
 - un iPad en vertical y conmuta entre los modos **.PrimaryHidden** y **.PrimaryOverlay**.
 - un iPhone 6 Plus en horizontal y conmuta entre los modos **.PrimaryHidden** y **.AllVisible**.

