Plataforma iOS

Sesión 7: Componentes para iPad y Aplicaciones Universales

Índice de contenidos

- **Componentes específicos para iPad:**
  - Split Views.
  - Popovers.

- Aplicaciones Universales en iOS.

Componentes para iPad

- Existen dos componentes específicos para iPad:
  - **Vistas divididas (Split View).**
  - Ventanas emergentes (Popovers).

¿Qué es un Split View?

- Tipo de vista especialmente diseñada para iPad en modo horizontal.

- En modo **horizontal** se divide la pantalla en dos partes:
  - Vista principal o *Master view*: Es la vista de la izquierda.

    Uso para listados (vista de tabla).
  - Vista detalle o *Detail View*: Es la vista de la derecha.

    Uso para el detalle de la selección de la vista de la izquierda. (cualquier tipo de vista).

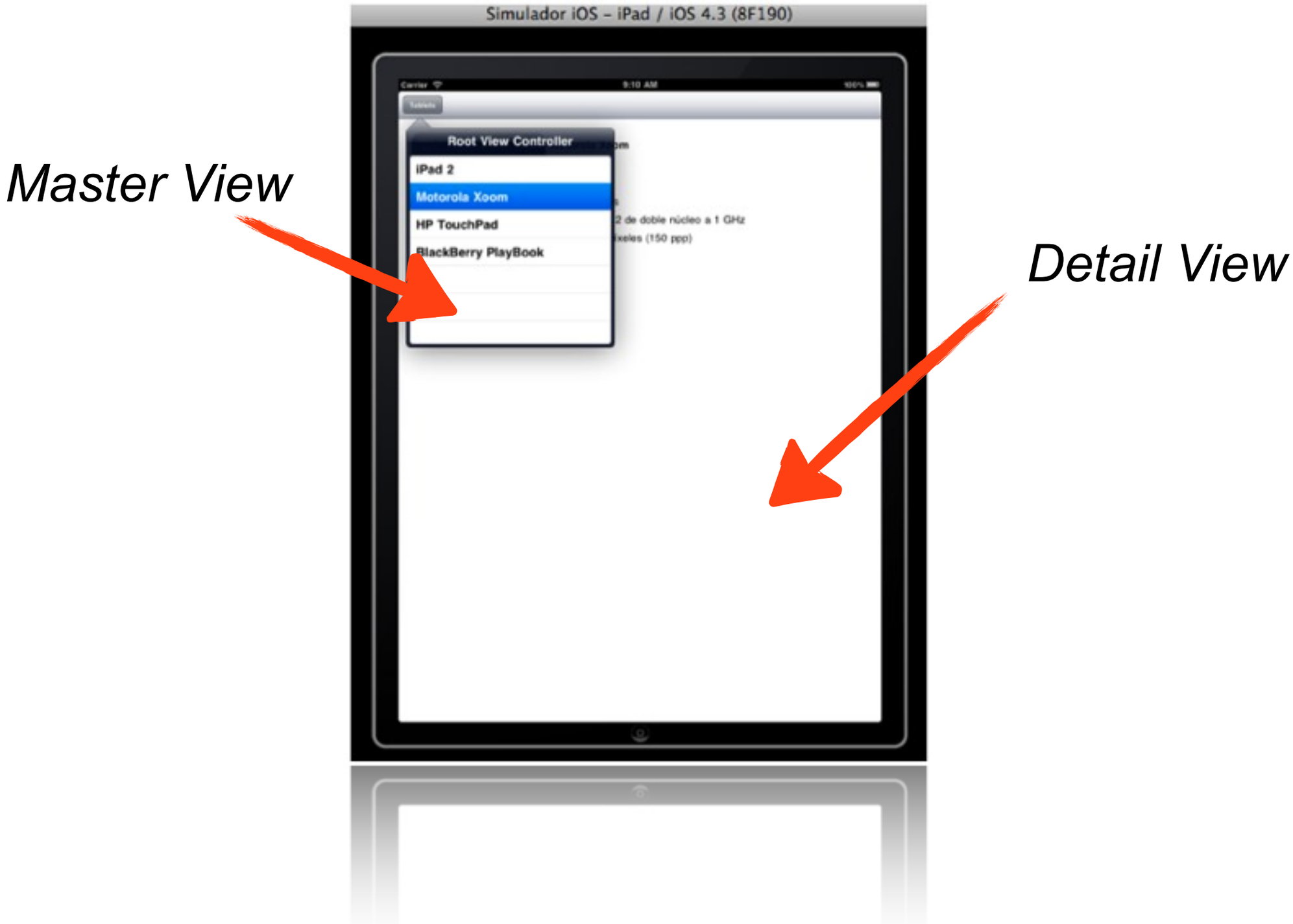
- En modo **vertical** se vería únicamente la vista de detalle y la vista principal aparecerá en forma de *Popover* desde la parte superior-izquierda.

Split View vista horizontal

Master View

Detail View

Split View vista vertical



Usos del Split View (1)

- Listados y detalle
  - La más habitual.
  - En la parte izquierda se muestra un listado de objetos.
  - En la parte derecha se muestran los detalles del objeto seleccionado.
  - Usado en pantallas de configuraciones entre otros.

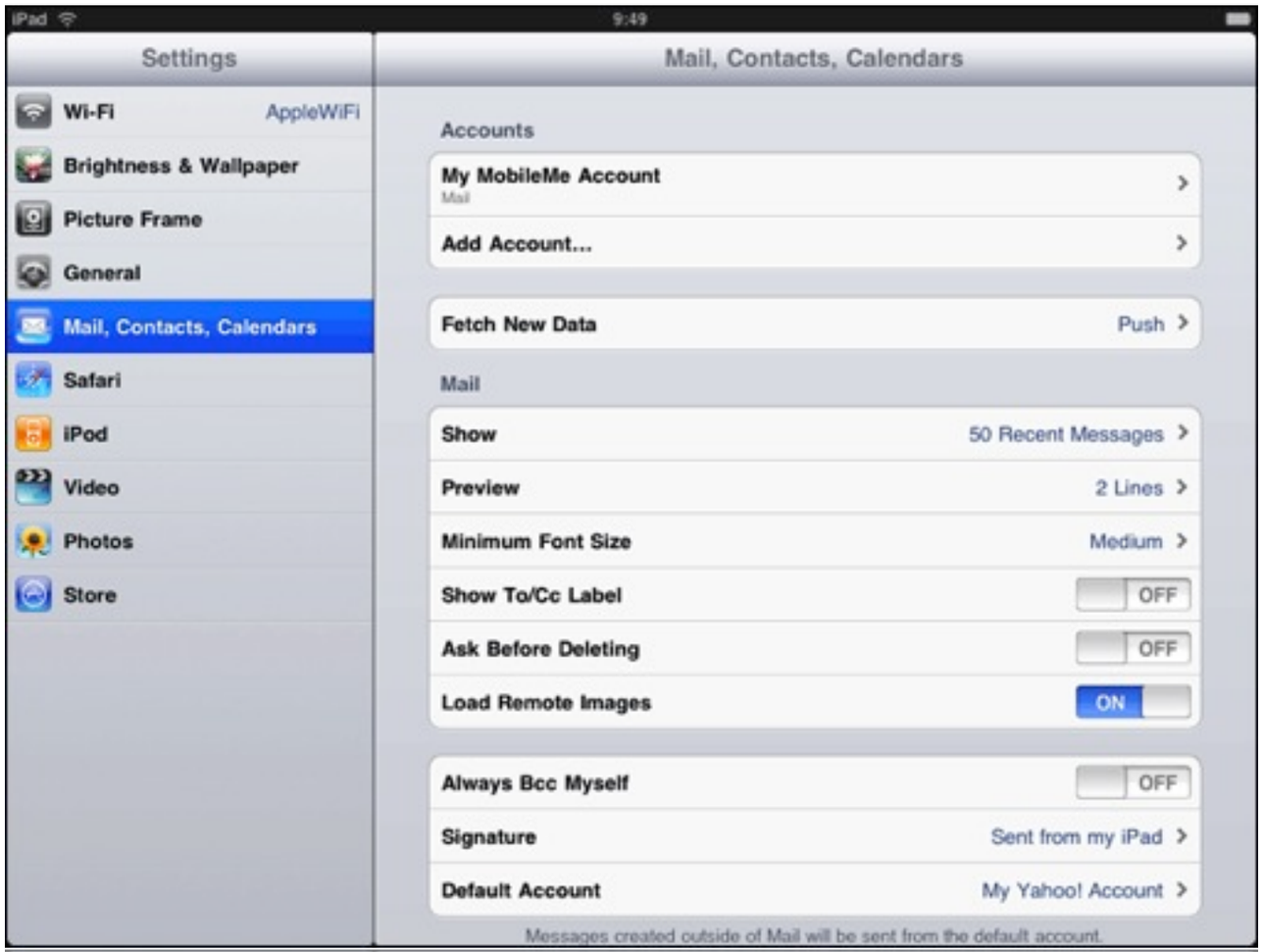
- Dos vistas independientes (sin comunicación directa entre ambas).

Usos del Split View (2)

- Listados y detalle:

Usos del Split View (3)

- Listados y detalle, pantallas de configuraciones:



Usos del Split View (4)

- Vistas independientes: la vista de la derecha no cambia según la selección de la izquierda:

Programando un *Split View (1)*

- Usando una versión de Xcode anterior 4.2:
  - Método tradicional.
  - Crear la controladora de forma programada y añadirle las vistas.
  - Programar la vista vertical y popover.
- Usando una versión de Xcode a partir de 4.2:
  - Creando un nuevo proyecto usando la plantilla "*Master Detail Application*".

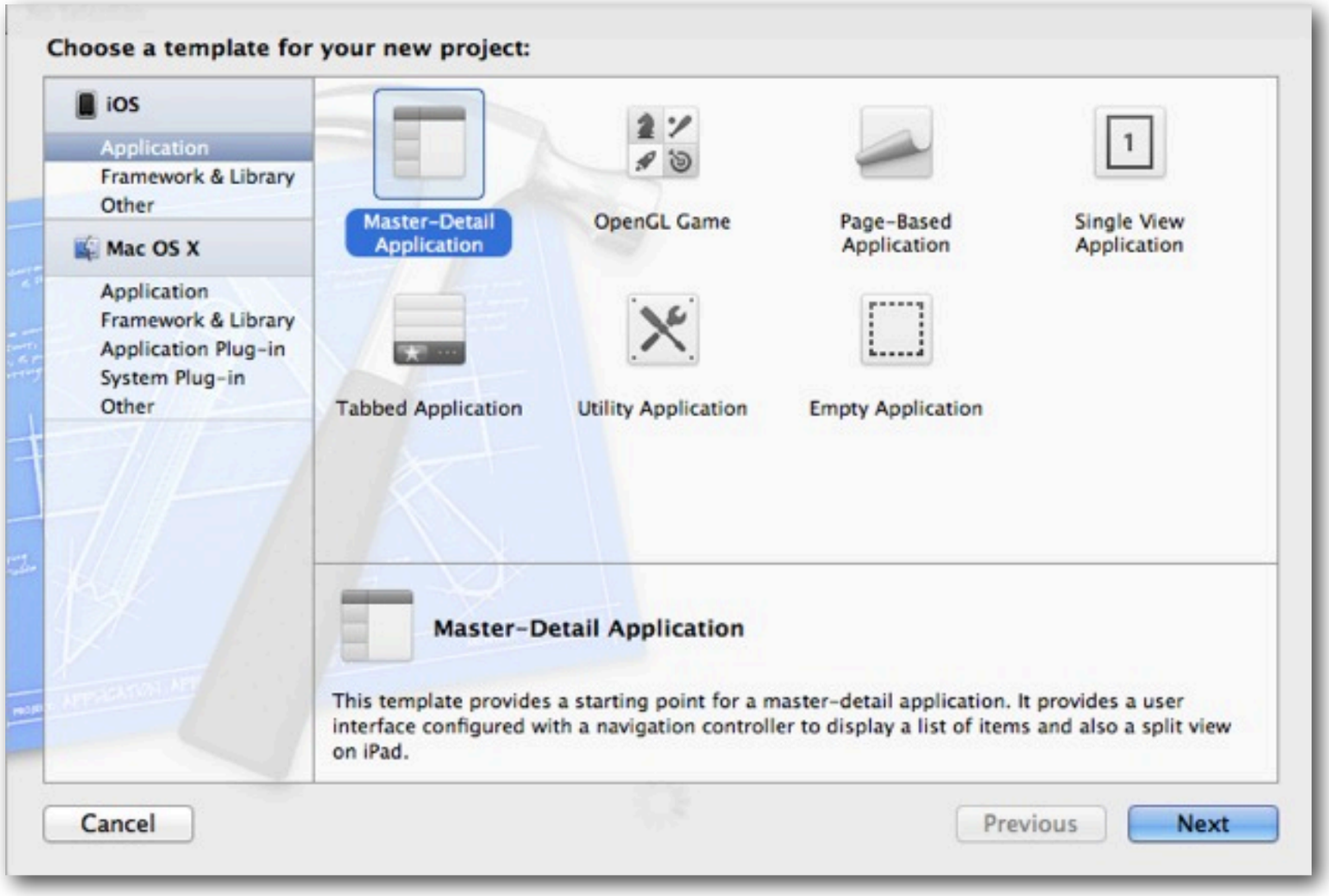
El *Split View* siempre debe de cargarse en el arranque de la aplicación, desde el controlador raíz, no se puede cargar dentro de una vista intermedia. Además sólo puede haber uno por aplicación.

Programando un *Split View (2)*

- Método tradicional (anterior a Xcode 4.2):
  - Diseñamos las dos controladoras con sus vistas por separado.
  - Creamos un objeto de tipo `UISplitViewController`.
  - Asignamos al objeto `UISplitViewController` las dos controladores creadas anteriormente.
  - Añadimos el objeto `UISplitViewController` a la ventana principal del proyecto en el arranque (método `didFinishWithLaunchOptions`).

Programando un *Split View (3)*

- Método con plantilla (a partir de Xcode 4.2):
  - Creamos un nuevo proyecto usando la plantilla de "*Master Detail Application*".



Programando un *Split View (4)*

- Programando el modo vertical:
  - En el método con **plantilla**, la programación de la vista vertical la evitamos.

    Esta incluye la creación del *Popover* con la vista principal.

  - En el método **tradicional** deberemos programar el evento de la vista vertical que muestra el *Popover* (si la vamos a utilizar en nuestra aplicación).

- La comunicación entre la vista principal y la vista detalle se debe de realizar mediante protocolos (clases delegadas).

Programando un *Split View (5)*

- Split View en una aplicación universal:
  - A partir de Xcode 4.2 podremos generar la estructura del proyecto para una aplicación Universal directamente desde la configuración.

    En este caso el controlador de *Split View* se reemplazará por un *Navigation Controller* común para el iPhone:

    Primera vista del Navigation Controller: Principal (*Master view*).

    Segunda vista del Navigation Controller: Detalle (*Detail view*).

Componentes para iPad

- Existen dos componentes específicos para iPad:
  - Vistas divididas (Split View).
  - **Ventanas emergentes (Popovers).**

¿Qué es un Popover?

- Ventana emergente (similar a un *popup* en Web).
- Permite un mayor aprovechamiento de la pantalla en un iPad y mejora la usabilidad de la aplicación.
- Puede contener casi todos los tipos de vistas disponibles.
- Puede aparecer en cualquier parte de la pantalla:
  - Desde un botón contenido en una barra de navegación o una barra de herramientas (*tool bar*).
  - Desde cualquier otra parte especificando el punto exacto al crearlo (*frame*).
- Se debe de cerrar cuando el usuario toca cualquier parte de la pantalla fuera del popover o cuando se llama al evento que lo ha mostrado.
- Según la guía de estilo de Apple no pueden existir dos popovers al mismo tiempo en pantalla -> Motivo de rechazo de la app.

Usos de un Popover (1)

- Mostrar listados.
  - Dentro de un *Split View Controller* en modo vertical.

- Mostrar cualquier otro tipo de información, formularios, otros listados (tablas), imágenes, objetos de tipo *action sheet*, etc.

- Ejemplos de uso...

Usos de un Popover (2)

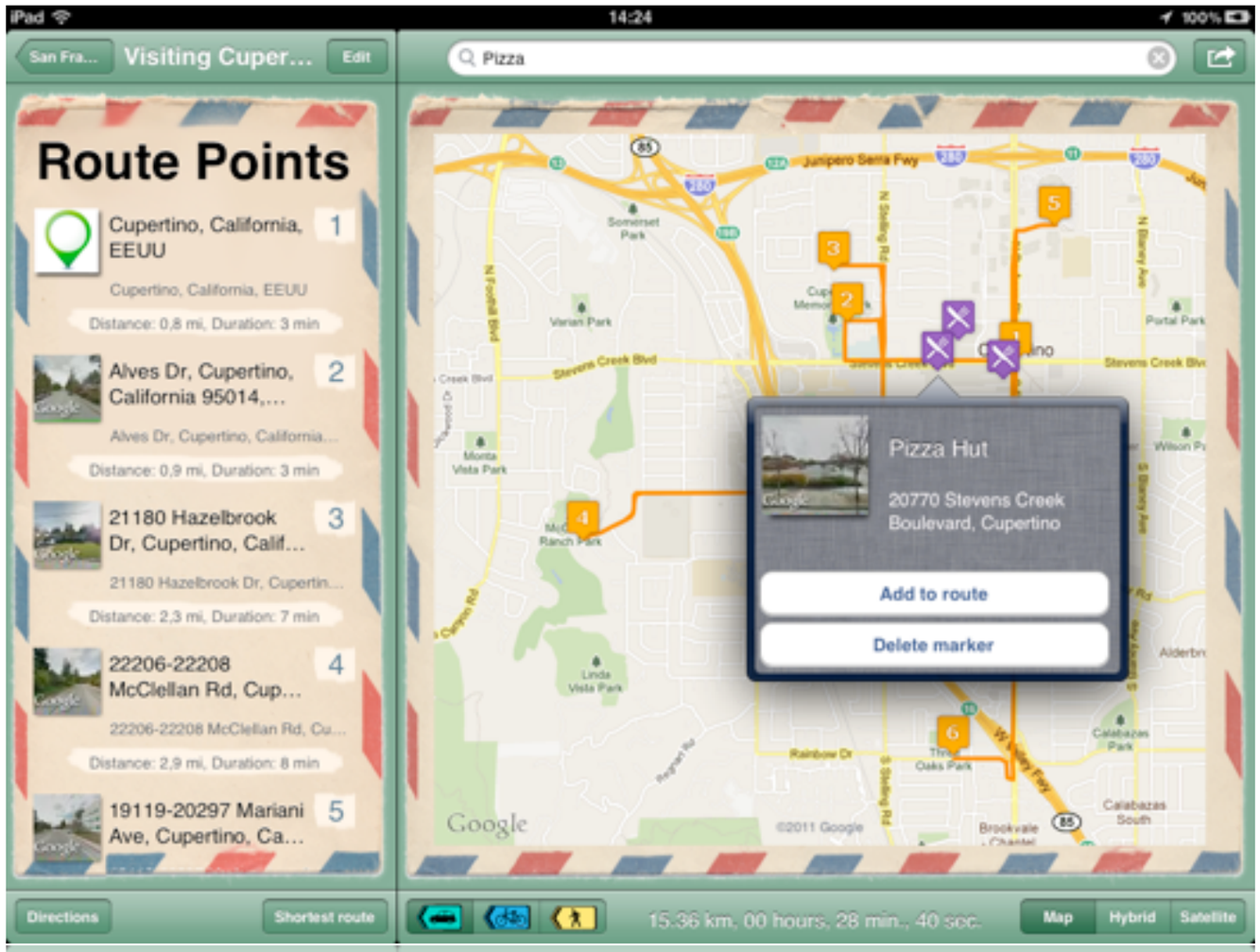
- Listado de objetos seleccionables. Aplicación Mail.

Usos de un Popover (3)

- Con un formulario de opciones. Aplicación de audio.

Usos de un Popover (4)

- En cualquier parte de la pantalla con información adicional. Aplicación de rutas.



Programando un Popover (1)

- Pasos para programar un Popover:
  - 1) Añadir el protocolo `UIPopoverControllerDelegate` a la declaración de la clase (*fichero .h*).
  - 2) Definir un objeto de tipo `UIPopoverController` en la definición de la clase.
  - 3) Crear en memoria el Popover *(normalmente en algún método de inicialización de la vista, viewDidLoad...):*

```
UIPopoverController *miPopover = [[UIPopoverController alloc] init];
```

Programando un Popover (2)

- 4) Donde queramos mostrar el Popover:
  - a) Comprobar si el popover está mostrándose en pantalla.

```
if (miPopover.popoverVisible)...
```

  - b) Creamos en memoria la vista que queremos mostrar dentro del Popover.

```
MiVistaViewController *miVistaEnPopover =
                    [[MiVistaViewController alloc] init.....];
```

- c) Asignamos al Popover la vista que hemos creado:

```
[miPopover initWithContentViewController: miVistaEnPopover];
```

- d) Mostramos el Popover:

```
[miPopover presentPopoverFromXXX: miObj
        permitedArrowDirections: UIPopoverArrowDirectionAny
                       animated: YES];
```

Programando un Popover (3)

- Para cerrar el Popover:

```
[miPopover dismissPopoverAnimated:YES];
```

(*) Los Popover son de los pocos objetos en iOS que no se pueden crear usando el Interface Builder de Xcode.

Índice de contenidos

- Componentes específicos para iPad:
  - Split Views.
  - Popovers.

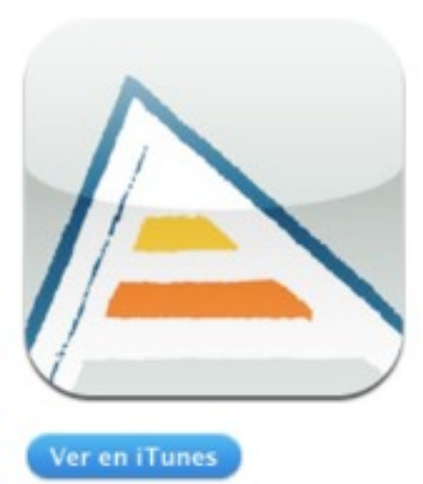
- **Aplicaciones Universales en iOS.**

Aplicaciones Universales (1)

- Aparecen a partir de iOS 3.2 (junto con la aparición del iPad).
- **Universal** = App compatible y adaptada a todos los dispositivos iOS que existen actualmente en el mercado (iPhone/iPad).
- Ventajas para el desarrollador:
  - Se programa la mayoría del código una sola vez.
  - Actualizaciones rápidas: sólo deberemos actualizar un proyecto XCode.
  - Se instala en todos los dispositivos iOS el mismo binario = más atractivo al usuario final = ¿más opciones de ventas/descargas?

Aplicaciones Universales (2)

- Desventajas para el desarrollador:
  - El código se complica algo más: sentencias condicionales para distinguir entre tipos de dispositivos, varias vistas por controlador, etc.
  - Actualizaciones algo más costosas: deberemos “tocar” más código.
  - Sólo se desarrolla una app = menos visibilidad en App Store = ¿posible disminución de ventas de aplicaciones?

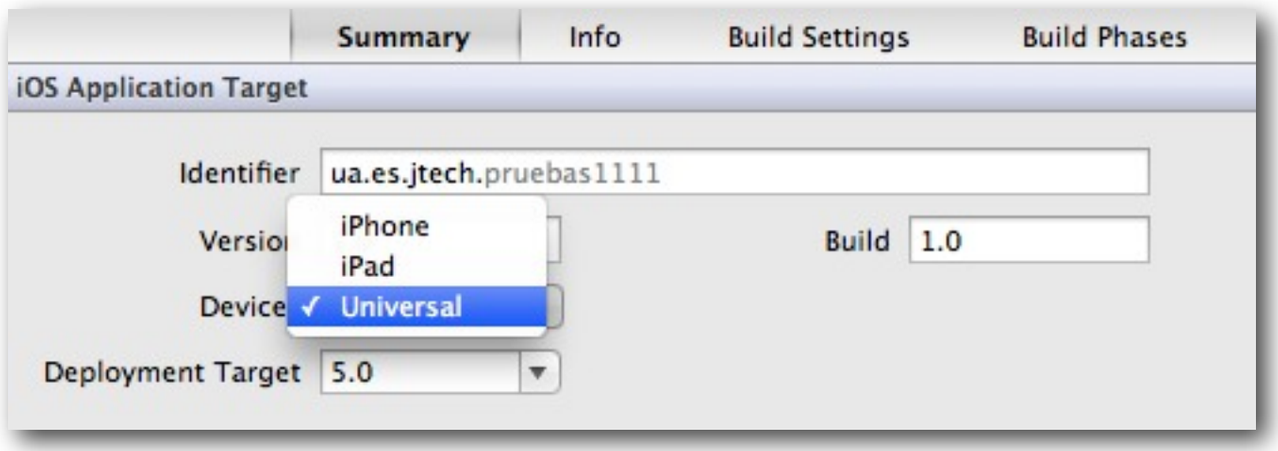


Programando una app Universal (1)

- A partir de Xcode 4.2 (iOS 5) podemos crear proyectos usando plantillas para aplicaciones universales.

- Normalmente distinguiremos únicamente entre dispositivos iPhone e iPad, no tendremos en cuenta Retina Display (iPhone 4, 4S), esto es más para aplicaciones de juegos.

- Pasos para programar una app Universal:
  - **1)** Modificar las especificaciones del proyecto:



Programando una app Universal (2)

- **2)** Diseñamos las vistas de iPhone y de iPad por separado (dos ficheros de Interface Builder, XIB distintos).

```
MiVistaViewController_iPhone.xib
MiVistaViewController_iPad.xib
```

- **3)** Implementamos la controladora para las vistas (**sólo una**):

```
MiVistaViewController.h
MiVistaViewController.m
```

Programando una app Universal (3)

- **4)** En donde vayamos a crear la controladora `MiVistaViewController` deberemos iniciarla con la vista de iPhone si estamos usando un iPhone o de iPad si estamos usando un iPad. Esto lo haremos usando el singleton `UIDevice:`

```
if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPhone) {

    self.viewController = [[ViewController alloc]
                           initWithNibName:@"ViewController_iPhone" bundle:nil];
}
else
{
    self.viewController = [[ViewController alloc]
                           initWithNibName:@"ViewController_iPad" bundle:nil];
}
```

Programando una app Universal (4)

- **5)** Iconos. En una aplicación Universal deberemos añadir al proyecto los iconos en las dimensiones correctas para iPad, iPhone 4 retina display, iPad retina...

- **6)** Lo mismo que el punto 5 deberemos hacer para las imágenes *default* (de entrada a la aplicación). Estas imágenes se arrastran a la pantalla de configuración del proyecto y deben de cumplir unas medidas y extensiones concretas.

(*) No deberemos realizar ningún cambio para adaptar la aplicación a la App Store. Esta aparecerá como "Universal" automáticamente.

¿Preguntas?