



POLITÉCNICA

ETSIT
UPM

dit
UPM

Obsoleto

Xcode 6 y iOS 8 introducen Layout Adaptativo

Desarrollo de Apps para iOS

Aplicaciones Universales

IWEB,LSWC 2013-2014

Santiago Pavón

ver: 2012.12.13 p5

Objetivo

Adaptar nuestras aplicaciones para que ejecuten tanto en un iPhone y como en un iPad.

Alternativas

- Una única aplicación *Universal* que se ejecuta tanto en iPhone como en iPad.
 - Crear dos aplicaciones distintas pero que comparten el mismo código.
 - Realizar dos desarrollos totalmente independientes.
- Para todas estas opciones, el modelo (La M del MVC) no debería cambiar.

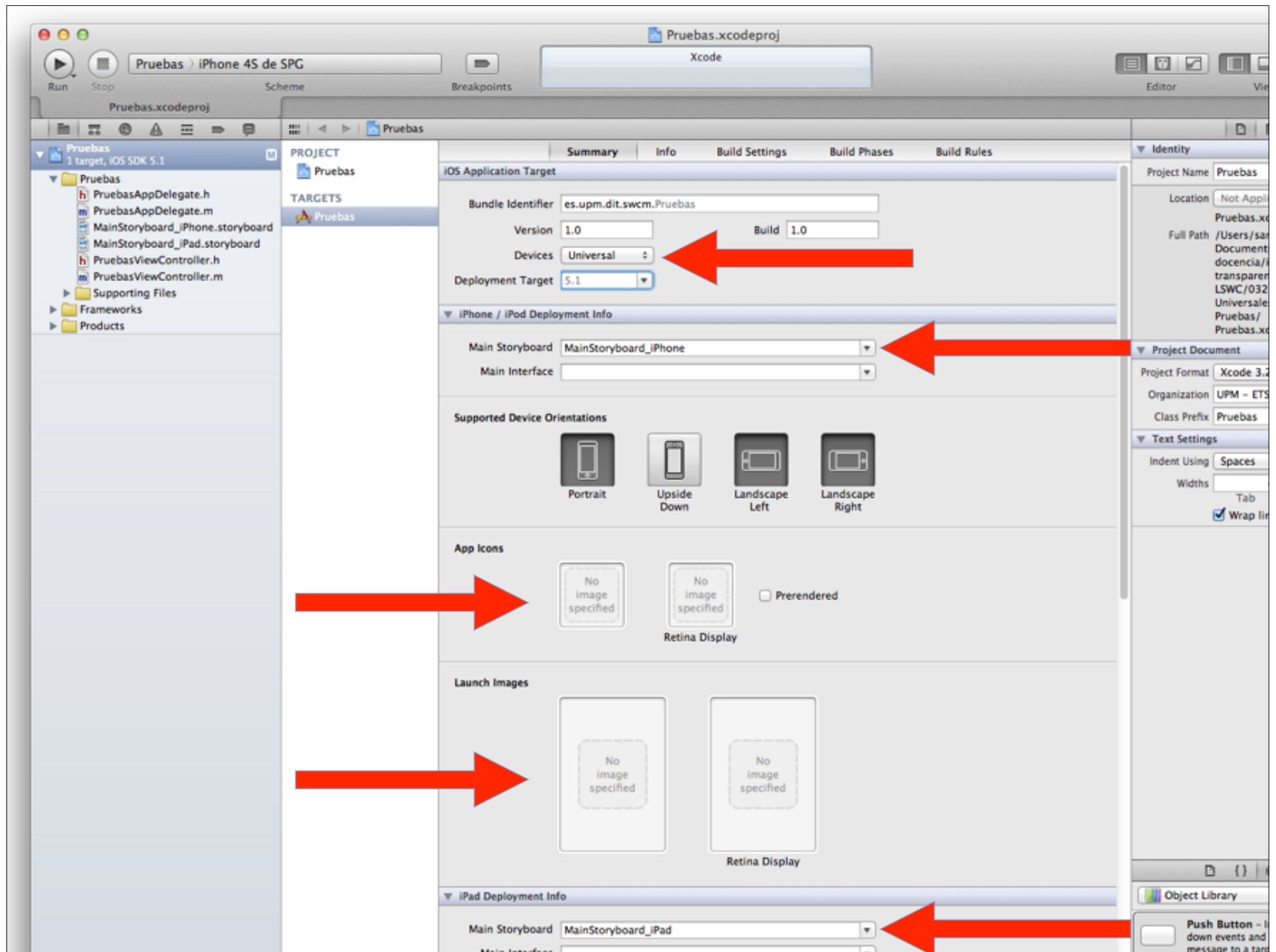
Diferencias iPhone -iPad

- Pantallas de diferente tamaño.
 - En el iPad cabe más información.
 - Debemos adaptar el tipo de navegación para cada dispositivo.
- Disponemos de más controladores en un iPad:
 - Split View Controller, Popover Controller
- En un iPad:
 - hay nuevas formas de presentar pantallas modales
 - hay más tipos de transiciones.

● . . .

Cambios a Realizar

- Cambiar la configuración del proyecto.
 - iOS Application Target - Devices
 - iPhone, iPad o Universal
 - iPhone/iPod y iPad Deployment Info
 - Crear un storyboard para iPhone y otro para iPad
 - Crear iconos y splash imágenes para cada tipo de dispositivo
- Refactorizar el código.
- Adaptar otros recursos.
- . . .



Adaptar Código

- En aplicaciones universales o que compartan código:
 - Refactorizar para no repetir código.
- Para soportar dispositivos con distintas versiones de iOS, y por tanto distintos APIs.
 - Preguntar si un objeto responde a un método determinado

```
[obj respondToSelector:@selector(metodo) ];
```
- Para conocer el tamaño de la pantalla

```
CGRect rect = [[UIScreen mainScreen] bounds];
```
- Para saber que tipo de dispositivo tenemos:

```
if (UI_USER_INTERFACE_IDIOM()==UIUserInterfaceIdiomPad)
```
- Para saber si mi controlador está embebido en un SplitVC

```
if (self.splitViewController) {}
```

- Un iPad puede mostrar en pantalla varios VC simultáneamente.
 - Para saber si un VC está visible (en pantalla) o no, se puede consultar el valor de la propiedad **window**.
 - Si el VC no está visible valdrá **nil**.
- . . .

Ejemplo de Refactorización

- Adaptar un método para que dependiendo del tipo de dispositivo, iPhone o iPad, se configure una propiedad o se dispare un segue.
 - Usando el IDIOM (UIDevice.h)

```
if (UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPad)
```

```
- (void)nextStep
{
    if (UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPad) {
        UIViewController *vc =
            [self.splitViewController.viewControllers lastObject];
        vc.msg = @"Hello";
    } else {
        [self.performSegueWithIdentifier:@"ShowMsg" sender:self];
    }
}
```

Ejemplo de Refactorización

- Usar herencia:
 - Derivar de una clase base una versión para iPad y otra para iPhone.
 - La clase base contiene todo el código común.
 - Las clases derivadas sobrescriben los métodos que son diferentes para ellas.
 - Desde los storyboards del iPhone y del iPad se hará referencia a la clase derivada que corresponda.

