ETSIT UPM

dit UPM

Desarrollo de Apps para iOS

Algunas `UIViews`

IWEB,LSWC 2013-2014
Santiago Pavón

ver: 2012.10.22 p1

UIViewImage

UIImageView

- Es una `UIView` en la que se muestra una imagen o una secuencia animada de imágenes.
- Puede crearse con el Interface Builder, o mediante código.

```
UIImage *img = [UIImage imageNamed:@"foto.jpg"];
UIImageView *iv = [[UIImageView alloc] initWithImage:img];
```

API

- Propiedades:

```
image
highlightedImage
animationImages
highlightedAnimationImages
animationDuration
contentMode
. . .
```

- Métodos:

```
startAnimating
stopAnimating
. . .
```

¿Cómo crear una UIImage?

```
UIImage *img = [UIImage imageNamed:@“a.jpg”];

NSStriing * path = ???;
UIImage *img = [UIImage imageWithContentsOfFile:path];


NSData * dat = ???;
UIImage *img = [[UIImage alloc] initWithData:dat];
```

UISegmentedControl

UISegmentedControl

- Control horizontal de botones.
- Sólo puede selecionarse un botón.
- Puede tener un título, NSString o UIImage.
- Puede presentarse con diferentes estilos: plano, como una barra, . . .
- Los botones pueden mostrar un NSString o una UIImage.
- Usa Target/Action para controlar sus cambios.

UIActionSheet

UIActionSheet

- Aparecen desde la parte inferior de la pantalla (iPhone) o como un popover (iPad).
- Se muestra para pedirle al usuario que tome una decisión.
- Normalmente hay varias alternativas a elegir.

Introducción a su Uso

- Método de inicialización:

```
- (id)initWithTitle:(NSString *)title
          delegate:(id <UIActionSheetDelegate>)delegate
          cancelButtonTitle:(NSString *)cancelButtonTitle
          destructiveButtonTitle:(NSString *)destructiveButtonTitle
          otherButtonTitles:(NSString *)otherButtonTitles, ...;
```

- Una vez creada pueden añadirse más botones con:

```
- (NSInteger)addButtonWithTitle:(NSString *)title;
```

- Para mostrarla se recomienda usar:
  - En un iPhone:

```
- (void)showInView:(UIView *)view;
- (void)showFromToolbar:(UIToolbar *)view;
- (void)showFromTabBar:(UITabBar *)view;
```

  - En un iPad:

```
- (void)showFromRect:(CGRect)rect
                inView:(UIView *)view
              animated:(BOOL)animated;
- (void)showFromBarButtonItem:(UIBarButtonItem *)item
                     animated:(BOOL)animated;
```

- En un iPad:
  - la action sheet es un popover.
    - No se muestra el botón cancel.
      - La action sheet desaparece al tocar fuera ella.
  - Pero si la action sheet se muestra dentro de un popover, entonces se comporta como si fuera mostrada en un iPhone.
    - En este caso, mostrarla usandoc `showInView:`.
    - La action sheet se comporta de forma modal: el popover no puede cerrarse mientras la action sheet siga presente.

- Para saber que botón pulsó el usuario debe implementarse alguno de los métodos del protocolo delegado `UIActionSheetDelegate`.

  - `actionSheet:clickedButtonAtIndex:`
  - `actionSheet:willDismissWithButtonIndex:`
  - `actionSheet:didDismissWithButtonIndex:`
  - `actionSheetCancel:`

- Para saber cual es el índice de los botones usar los siguientes metodo y propiedades de la action sheet

```
cancelButtonIndex
destructiveButtonIndex
firstOtherButtonIndex
numberOfButtons
- buttonTitleAtIndex:
```

- Si la action sheet se mostró como un popover en un iPad, y el popover desapareció porque se pulsó fuera de él, entonces el índice del botón es -1.

```
    - (IBAction)pulsamePressed:(id)sender {
     UIActionSheet *sheet = [[UIActionSheet alloc] initWithTitle:@"¿Está seguro?"
                                                        delegate:self
                                               cancelButtonTitle:@"No"
                                          destructiveButtonTitle:@"ñaka ñaka"
                                               otherButtonTitles:@"Pregúntame mañana",
                                                                 @"Pregúntame después",
                                                                 nil];
     [sheet showInView:self.view];
}

-(void)actionSheet:(UIActionSheet *)actionSheet
        clickedButtonAtIndex:(NSInteger)buttonIndex {
     if (buttonIndex == actionSheet.cancelButtonIndex) {
         return;
     }
     if (buttonIndex == actionSheet.destructiveButtonIndex) {
         NSLog(@"Hacer ñaka ñaka");
         return;
     }
     NSString * msg = [actionSheet buttonTitleAtIndex:buttonIndex];
     NSLog(@"Indeciso: %@", msg);
}
```

- Cuidado: En un iPad:
  - Si la action sheet se muestra desde un UIBarButtonItem, la toolbar de ese botón se añade a las pasthrouhgViews del popover.
    - Entonces puedo seguir pulsando el BarButtonItem y sacar más actions sheet simultaneamente.
    - O pulsar cualquier otro botón de la toolbar.
  - Para evitar este problema debemos programar algo. Por ejemplo, se puede deshabilitar la toolbar mientras la action sheet esté visible, asignando `NO` a su propiedad `userInteractionEnabled`.
    - Cuando desaparezca la action sheet se asignará YES.

```
- (void)didPresentActionSheet:(UIActionSheet *)actionSheet {
   self.toolbar.userInteractionEnabled = NO;
}
- (void)actionSheet:(UIActionSheet *)actionSheet
        didDismissWithButtonIndex:(NSInteger)buttonIndex {
   self.toolbar.userInteractionEnabled = YES;
}
```

UIAlertView

UIAlertView

- Aparece en mitad de la pantalla
- Hace una pregunta con una o dos respuestas.

Introducción a su Uso

- Método de inicialización:

```
- (id)initWithTitle:(NSString *)title
            message:(NSString *)message
           delegate:(id)delegate
  cancelButtonTitle:(NSString *)cancelButtonTitle
  otherButtonTitles:(NSString *)otherButtonTitles, ...
```

- Una vez creada pueden añadirse más botones con:

```
- (NSInteger)addButtonWithTitle:(NSString *)title;
```

- Para mostrar la alert view llamar al método:

```
-(void) show;
```

- Para saber que botón pulsó el usuario, implementarselguno de los métodos del delegado `UIAlertViewDelegate`.

```
– alertView:clickedButtonAtIndex:
– alertView:willDismissWithButtonIndex:
– alertView:didDismissWithButtonIndex:
– alertViewCancel:
```

- Para saber cual es el índice de los botones usar los siguientes métodos y propiedades

```
cancelButtonIndex
firstOtherButtonIndex
numberOfButtons
- buttonTitleAtIndex:
```

```
    - (IBAction)pulsamePressed:(id)sender {

    UIAlertView * alert = [[UIAlertView alloc]
          initWithTitle:@"Login"
                message:@"Para usar los servicios ofertados es necesario conectarse.
\n¿Qué cuenta desea usar para conectarse?"
               delegate:self
      cancelButtonTitle:@"Abandonar"
      otherButtonTitles:@"Facebook", @"Tweeter", nil];

    [alert show];
}

-(void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex
{
    if (buttonIndex == alertView.cancelButtonIndex) {
        return;
    }

    NSString * msg = [alertView buttonTitleAtIndex:buttonIndex];
    NSLog(@"Conectarse por %@", msg);
}
```

- Existen varios estilos de alertas que se especifican asignado el valor adecuado a la propiedad alertViewStyle
  - **`UIAlertViewStyleDefault`**
    - Una alerta estandar.
  - **`UIAlertViewStyleSecureTextInput`**
    - Alerta con un campo TextField seguro para introducir texto.
  - **`UIAlertViewStylePlainTextInput`**
    - Alerta con un campo TextField para introducir texto.
  - **`UIAlertViewStyleLoginAndPasswordInput`**
    - Alerta para introducir un login y un password.

- A los campos de texto se accede con el método:
  - `(UITextField *)`**`textFieldAtIndex:`**`(NSInteger)index:`

```
UIAlertView * alert = [[UIAlertView alloc]
      initWithTitle:@"Login"
            message:@"Introduzca su nombre:"
           delegate:self
  cancelButtonTitle:@"Abandonar"
  otherButtonTitles:nil];

alert.alertViewStyle = UIAlertViewStylePlainTextInput;
[alert show];
```

```
-(void)alertView:(UIAlertView *)alertView
       clickedButtonAtIndex:(NSInteger)buttonIndex
{
    if (buttonIndex == alertView.cancelButtonIndex) {
        return;
    }

    UITextField * tf = [alertView textFieldAtIndex:0];
    NSLog(@"Nombre = %@", tf.text);
}
```