

- Multiple MVCs
- Aggiungere un view controller
  - UIViewController
  - UINavigationController
  - UINavigationController
  - Segue
  - UITabBarController
    - Le viste di un UITabBarController

## Nota

La gestione di più viste significa avere più view-controllers che devono coordinare il flusso dell'applicazione.

## Ricorda

## Multiple MVCs

## Nota

- App Contatti
- App Calendario
- App Musica

- UINavigationController
- UITabBarController

## Aggiungere un view controller

1. trascinando un `UIViewController` dal palette degli oggetti alla storyboard.
2. Creando una sotto classe di `UIViewController` utilizzando il menu di creazione di un nuovo file.
3. Nell'identity inspector impostando una classe di `UIViewController` a nuova sotto-classe creata.

Metodo	Descrizione
<code>-(void)pushViewController:(UIViewController *)viewController animated:(BOOL)animated</code>	È utilizzato per mettere una lista in cima allo stack

Metodo	Descrizione
<code>(UIViewController *)popViewControllerAnimated:(BOOL)animated</code>	È utilizzato per rimuovere una vista dalla cima dello stack
<code>(NSArray *)popToViewController:(UIViewController *)viewController animated:(BOOL)animated</code>	Rimuove dalla cima dello stack tutti view controllers finché non viene trovato quello desiderato
<code>(NSArray *)popToRootViewControllerAnimated:(BOOL)animated</code>	Rimuove tutti i view controller dallo stack escludendo il view controller root

#### Nota

Tutti gli `UIViewController` presentano la proprietà `navigationController` che può essere usata per accedere al `UINavigationController` che la incorpora.

## UINavigationController

La vista per un navigation controller è solo un container per diverse altre viste. Alcuni esempi di navigation controller sono:

- Barra di navigazione
- Barra degli strumenti
- Vista con contenuti personalizzati

Quando ci si muove da un MVC ad un altro, il contenuto della vista personalizzata cambia, come cambierà anche la barra di navigazione e la barra degli strumenti.

## UINavigationController

La `UINavigationController` mostra:

- Un titolo, per il view controller corrente. Può essere imposta to mediante la proprietà `title` del `UIViewController` corrispondente
- Un bottone per tornare indietro (back button), che mostra il `title` del `UIViewController` precedente nel navigation stack
- Un array di oggetti `UIBarButtonItem` accessibile con la proprietà `navigationItem.rightBarButtonItem` del `UIViewController`

#### Nota

Un `UINavigationController` inizia mostrando nessun contenuto, poi quando la proprietà `rootViewController` è impostata il `UINavigationController` mostrerà le viste gestite dal view controller.

## Segues

Le segue sono responsabili di svolgere la transazione (attivate mediante delle segue o mediante codice) visiva tra 2 view controllers.

Tipi di segue
Push
Modal
Custom

Un view controller può svolgere azioni aggiuntive prima di performare una segue, che necessita il tempo di essere preparata, ovvero necessita di passare i dati importanti alla nuovo view controller.

Ogni view controller può sovrascrivere il metodo `(void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender` per prepararsi per la segue. Il metodo è chiamato prima he il nuovo view controller (`destinationViewController` della segue) è presentato.

*Esempio preparazione per una segue*

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    if([segue.identifier isEqualToString:@"GoToSecondView"]) {
        if([segue.destinationViewController isKindOfClass:[SecondViewController
class]]) {
            SecondViewController *sVC =
            (SecondViewController*)segue.destinationViewController; sVC.data = @"Some data";
        }
    }
}
```

*Instanziazione di un view controller programmaticamente*

Un view controller può essere istanzializzato a mano (questo può avvenire quando il tipo di destinazione del view controller viene risolto a tempo di esecuzione).

```
NSString *vc = @"SecondViewController"; UIViewController *controller = [self.storyboard
```

```
instantiateViewControllerWithIdentifier:vc];
```

Il view controller, dopo essere stato creato, viene inserito in cima al navigation stack con il metodo `pushViewController:animated:`

## UITabBarController

La classe `UITabBarController` che implementa un view controller specializzato che gestisce una selezione di interfacce radio-style. Un utilizzo tipico è quello delle multiple sections (es. App Orologio).

Le schede sono mostrate sul fondo della finestra e permettono di scegliere tra differenti modalità.

L'interfaccia di ogni scheda di una tab bar controller è associata a un view controller personalizzato, quando un utente seleziona una scheda specifica, la barra delle schede mostra la root view del corrispondente view controller, rimpiazzando ogni view precedente. I view controller sono assegnati sul tab bar controller impostando la proprietà `viewControllers` del `UITabBarController`.

### Le viste di un UITabBarController

I view controller incorporati definiscono l'aspetto della loro tab bar item corrispondente. Il titolo per la scheda che correntemente mostrata il proprio view controller può essere impostata utilizzando la proprietà `title` del `UITabController` incorporato.

#### Nota

L'aspetto del titolo e dell'icona delle schede della barra possono essere configurate nello storyboard. In alternativa è possibile farlo programmaticamente creando un'istanza di `UITabBarItem` per configurare immagine e titolo, ed impostare la proprietà `tabBarItem` del controller incorporato con l'oggetto creato.

#### Nota

Lo spazio dedicato al tab bar controller è limitato, quindi quando sono presenti più di 6 view controllers viene aggiunta in automatico la scheda "morte"