I.T.I.S."E.MATTEI"

ANNO SCOLASTICO 2015-2016

IVAN MOTASOV CLASSE 5E

INDIRIZZO INFORMATICA

# **SARplay**



SARplay
Un'app per iPad che mostra le deformazioni del terreno
sulla base di dati satellitari

## **INDICE**

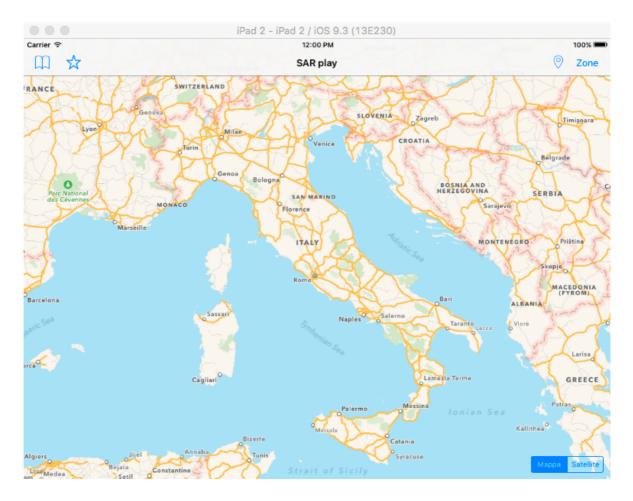
INTRODUZIONE	3
RACCOLTA, ELABORAZIONE E VISUALIZZAZIONE DEI DATI LATO WEB	5
METODOLOGIA DI LAVORO: BASECAMP	7
METODOLOGIA DI LAVORO: BITBUCKET	
LA MIA PARTE DI LAVORO	11
AMBIENTE DI LAVORO	12
I MENU DI SARPLAY	13
IL MENU DELLE ZONE	
STRUTTURA DELLA CLASSE ZONES VIEW CONTROLLER	
IL MENU DEI PREFERITI	17
INSERIRE UNA ZONA TRA I PREFERITI	18
STRUTTURA DELLA CLASSE BOOKMARKS VIEW CONTROLLER	19
I SEGUE	21
CONCLUSIONI	22
BIBLIOGRAFIA	23

### Introduzione

L'applicativo "SARplay" nasce grazie a due eventi che si sono svolti presso l'I.T.I.S. "E.Mattei" di Sondrio, durante l'anno scolastico 2015-2016:

- il primo riguarda la collaborazione della nostra scuola con la software house "Visuality Software srl", la quale, in cerca di una sede a Morbegno e di nuovo personale, si è offerta di istruire e trasmettere la sua esperienza nel mondo Apple verso gli studenti interessati;
- · il secondo evento vede coinvolto l'ex professore di Informatica Sabatino Bonanno, il quale, dopo aver insegnato per alcuni anni presso l'I.T.I.S. di Sondrio e il Liceo Scientifico "C. Donegani", è tornato a Napoli, dove attualmente lavora per il CNR (Consiglio Nazionale delle Ricerche), e si occupa di elaborazione di immagini satellitari, riguardanti la deformazione della crosta terreste nel tempo

(Istituto Rilevamento Elettromagnetico L'istituto IREA per il dell'Ambiente, dipartimento del CNR) ha il compito di raccogliere ed elaborare i dati dei satelliti, i quali vengono poi rappresentati su una mappa, in maniera che essa possa contenere tutti i punti analizzati. La mappa viene messa a disposizione chiedendo l'autorizzazione al CNR, ed è consultabile tramite un web server; tuttavia, la fluidità dell'interfaccia web dipende molto dalla connessione di cui si dispone ed inoltre è parecchio complicata da utilizzare; pertanto un utente medio non sarebbe in grado di farne un buon uso. L'unione di queste due problematiche ci ha spinti a pensare ad un'app per iPad semplice ed intuitiva, da realizzare insieme alla Visuality Software e con la collaborazione dell'IREA



L'app è formata dai seguenti elementi:

- connettore, ovvero una classe che si interpone tra server e applicazione; il suo compito è quello di scaricare un file json dal server, tramite una richiesta http
- database, contenente tutti i punti scaricati dal server tramite il connettore.
   È formato dalle entità "point", "zone" e "deformation"
- mappa, contenuta nella view principale, nella quale saranno visibili i punti in base alla zona selezionata. È possibile scegliere due layout diversi (satellitare e mappa semplice)
- posizione corrente, in modo che la mappa visualizzi i punti di deformazione della zona in cui si trova l'utente
- menu delle zone disponibili, costituito da una tabella contenente tutte le zone messe a disposizione dal connettore
- menu dei preferiti, costituito anch'esso da una tabella contenente tutte le zone visitate maggiormente dall'utente. Per poter inserire una zona tra i preferiti, si può utilizzare il bottone apposito, raffigurato da una stella

### Raccolta, elaborazione e visualizzazione dei dati lato Web

Per poter visualizzare la deformazione del terreno nelle zone specificate e analizzate dal CNR, i dati devono essere raccolti, elaborati e visualizzati:



Dallo schema si possono notare le 5 fasi che il processo richiede:

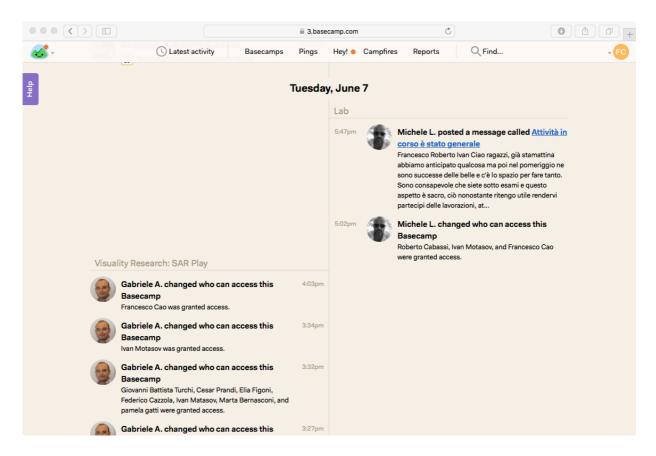
1- La fase di acquisizione dei dati è affidata al satellite il quale esegue i rilevamenti tramite l'uso di radar SAR. I sensori SAR sono associati a specifiche bande dello spettro elettromagnetico. Nelle applicazioni InSAR le bande comunemente utilizzate sono la banda L (frequenza 1-2 GHz, lunghezza d'onda ~24 cm), la banda C (frequenza 5-6 GHz, lunghezza d'onda ~6 cm) e la banda X (frequenza 8-12 GHz, lunghezza d'onda ~3 cm). Il principio di funzionamento di questi sensori è il seguente: un antenna trasmittente propaga nello spazio un' onda elettromagnetica che, incidendo sulla superficie terrestre, subisce un fenomeno di riflessione. Una parte del campo diffuso torna verso la stazione trasmittente, equipaggiata per la ricezione, dove vengono misurate le sue caratteristiche. Il dispositivo è in grado di individuare il bersaglio elettromagnetico (funzione di detecting) e, misurando il ritardo temporale tra l'istante di trasmissione e quello di ricezione, valutare la distanza a cui è posizionato, localizzandolo in modo preciso lungo la direzione di puntamento dell'antenna (direzione di range).

Il segnale radar relativo ad un bersaglio è caratterizzato da due valori: ampiezza e fase.

Questi valori permettono di realizzare due immagini. La fase in particolare racchiude l'informazione più importante ai fini delle applicazioni interferometriche.

- 2-Dopo aver raccolto i vari dati questi verranno inviati ai centri di trasmissione mediante l'uso di antenne satellitari. Queste faranno da intermediari con i server del CNR ai quali saranno inviate tutte le informazioni in formato stringa e ancora da elaborare.
- 3- Il compito dei server è quello di prendere le immagini satellitari ed elaborarle. L'elaborazione consiste nel rappresentare un immagine detta interferometria. L'interferometria è la misurazione delle variazioni della fase del segnale SAR tra due acquisizioni distinte. Una seconda elaborazione consiste invece nella trasformazione dell'interferometria in formato stringa, in modo da rendere semplice la gestione dell'immagine.
  - Questo intero processo, totalmente automatizzato tramite l'uso di programmi in C, è realizzato tramite l'utilizzo di una architettura hardware per elaborazioni in parallelo, chiamata CUDA (Compute Unified Device Architecture), sviluppata da NVIDIA. Essa permette di raggiungere alte prestazioni di computing grazie alla potenza di calcolo delle GPU. Nel caso analizzato vengono utilizzate schede video tesla k20.
- 4- I server oltre che ad occuparsi dell'elaborazione dei dati svolgono anche ruolo di WEB Server e si occupano di gestire una porzione del sito del CNR con il compito di mostrare i dati raccolti.
- 5- Tramite internet e con l'utilizzo di un browser si è in grado di collegarsi col CNR e visualizzare così su una mappa i dati sotto forma di punti. Tramite diverse gradazioni di colore si può capire il movimento del terreno.

### METODOLOGIA DI LAVORO: BASECAMP



Dal momento che SARplay nasce come progetto di gruppo, sin da subito c'è stata la necessità di trovare un modo per rimanere in contatto tra tutti e di poter condividere tra tutti il progetto di lavoro. Oltre ai numerosi incontri tenuti durante l'anno, sia con i membri della Visuality, sia solo tra noi ragazzi, rimaneva comunque il problema della condivisione del materiale, con mezzi che non fossero le e-mail. Gli sviluppatori di software utilizzano strumenti come LinkedIN per poter rimanere in contatto facilmente e per poter condividere il materiale tramite le apposite repository. Nel nostro caso, la Visuality software usa da tempo Basecamp (<a href="https://3.basecamp.org/">https://3.basecamp.org/</a>), pertanto ha creato uno spazio apposito per il progetto (un Basecamp, appunto).

Le principali funzionalità di Basecamp sono:

- Ping: sono dei messaggi tra due o più utenti, molto comodi per chiarire, specificare o chiedere concetti in maniera immediata; il funzionamento è lo stesso di ogni programma di chat;
- Hey: è una schermata contenente tutte le notifiche;
- Messages: è una chat tra tutti i membri del gruppo;
- To-dos: è una schermata che contiene tutte le cose da fare: ogni utente può creare dei to-do, e delegare ad un altro utente il compito; una volta terminato il compito, è possibile archiviarlo;
- Schedule: contiene tutte le scadenze dei to-dos. È possibile visualizzare tutti i to-dos di tutti i mesi futuri, oltre a quelli già scaduti;
- Check-ins: sono delle domande programmate che vengono inviate all'utente costantemente. Una domanda può essere "Sei bloccato su qualcosa?", oppure "Su cosa stai lavorando?"; in base alla risposta inviata, si possono ricevere chiarimenti, oppure si può organizzare ancora meglio il lavoro rimanente;
- Docs & files: è uno spazio nel quale vengono caricati file utili per il progetto corrente;

### METODOLOGIA DI LAVORO: BITBUCKET

Il progetto vero e proprio contenente il codice non viene caricato su basecamp, in quanto è scomodo e quasi impossibile tenere traccia di tutti gli aggiornamenti: per questo Visuality usa BitBucket (<a href="http://bitbucket.org">http://bitbucket.org</a>) come servizio di source control & versioning. BitBucket, fratello di GitHub, è un repository hosting basato su git. Git è in assoluto il sistema di versioning più utilizzato da parecchio tempo.

Per versioning si intende un sistema in grado di tenere traccia di tutti i cambiamenti avvenuti ad uno o più files nel tempo, così da poterne recuperare una versione precedente in qualunque momento, capire come è mutato un progetto nel tempo, sapere chi ha modificato qualcosa e quando. Lavorare appoggiandosi a git significa che qualsiasi errore o malfunzionamento introdotto dal programmatore può essere ripristinato in pochi secondi.

Per usare questo sistema si può scegliere di usare il terminale e dare i vari comandi operativi tramite CLI oppure di appoggiarsi a vari programmi che mettono a disposizione una GUI di facile utilizzo.

Git, considera i propri dati come una serie di istantanee (snapshot) di un mini filesystem. Ogni volta che l'utente effettua un commit, o salva lo stato del proprio progetto, fondamentalmente fa un'immagine di tutti i file presenti in quel momento, salvando un riferimento allo snapshot. Se alcuni file non sono stati modificati, Git non li clona ma crea un collegamento agli stessi file della versione precedente.

Un progetto Git è composto dai seguenti elementi:

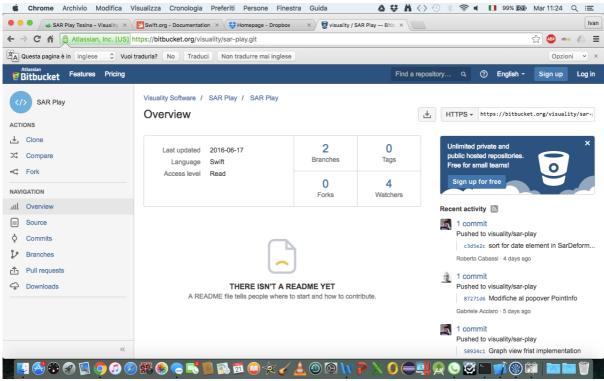
- Working dir o directory di lavoro che contiene i file appartenenti alla versione corrente del progetto sulla quale l'utente sta lavorando.
- Index o Stage che contiene i file in transito, cioè quelli candidati ad essere committati.
- · Head che contiene gli ultimi file committati.

É possibile inizializzare un nuovo progetto Git in due modi:

- Definire un nostro progetto preesistente come GIT Repository.
- Clonare un repository Git esistente da un altro server.

Oltre alle funzionalità base qui elencate, git offre moltissime altre features; per maggiori informazioni si rimanda al sito di supporto di git (<a href="https://www.git-scm.org">https://www.git-scm.org</a>).

Ecco una schermata di come si presenta l'ambiente bitbucket, contenente il progetto SARplay:



A sinistra dell'ambiente di lavoro, vi sono tutte le funzioni disponibili, come ad esempio il "clone", mentre nella parte di destra vi è in alto il link, per poter scaricare il progetto, e nella parte inferiore vi sono tutte le attività recenti legate al nostro progetto

### La mia parte di lavoro

Dal momento che l'applicazione risultava essere parecchio complessa, è stato pensata, giustamente, un'equa suddivisione del lavoro, in maniera che il lavoro venisse equamente distribuito e che ciascun gruppo di lavoro potesse specializzarsi su una parte dell'applicazione in particolare. Le parti dell'app da sviluppare sono state le seguenti:

- Connettore, di cui si è occupato Roberto Cabassi
- Mappa, di cui si è occupato Francesco Cao
- Database, di cui si è occupato Elia Figoni
- Parte dei menu, di cui mi sono occupato io

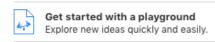
### Di seguitò si analizzeranno:

- · Ambiente di lavoro
- Classi e metodi utilizzati, con relativa spiegazione

### AMBIENTE DI LAVORO



Version 7.3.1 (7D1014)







Come ambiente di sviluppo per iOS, Apple mette mondo disposizione Xcode, strumento comodo е disponibile sull'AppStore in maniera gratuita. Il grande vantaggio di Xcode, secondo il sig. Longhi, della Visuality, sta nel fatto che tutte le applicazioni per il mondo Apple sviluppate siano solo tramite quest'ultimo. perfino quelle rilasciate dalla casa madre; ciò implica che per Apple stessa sia

utile avere un tool veloce, fluido, comodo da usare ed abbastanza intuitivo; lo stesso non vale per altre piattaforme, come .NET ad esempio, piattaforma con la quale il sig.Longhi ha lavorato per molti anni prima di passare ad Apple.

Per poter testare l'applicazione, Xcode permette di usare il dispositivo reale, in maniera che in fase di testing si possano vedere le reali prestazioni dell'app, l'utilizzo della memoria, della batteria e dei livelli di rete, oppure si può ricorrere ad un simulatore, il quale emula un dispositivo reale a scelta tra i vari iPad, iPod o iPhone che supportano la versione 9 di iOS

# iPad 2 - iPad 2 / iOS 9.3 (13E230) Carrier ♥ SAR play SAR play Roma Sondrio Milano Roma Otranto Barcelona Barcelona Sassasii Naples Salarno Larisa

### I MENU DI SARPLAY

Tra le varie features di SARplay, vi è la possibilità di scegliere la zona su cui operare in maniera manuale, oltre alla possibilità di poter contrassegnare le zone visitate maggiormente come "preferiti" e di vederle comparire in un menu a parte. Scegliendo una zona, si viene rimandati sulla regione di mappa che ci interessa, senza dover inserire coordinate o riferimenti geografici ulteriori.

I menu sono formati dai seguenti elementi:

- Bottone
- Tableview
- Segue

Il bottone ha il compito di fare da collegamento tra la mappa e il nostro menu: infatti, cliccandovi sopra, si aprirà la nostra tableView, contenente l'elenco delle zone; ciò che si interpone tra bottone e tableView, si chiama segue; nelle pagine seguenti verrà illustrato tutto il meccanismo che si cela dietro a questi tre elementi

### IL MENU DELLE ZONE

```
//
// ZonesViewController.swift
// SARplay
//
// Created by Michele Longhi on 24/02/16.
// Copyright © 2016 Visuality srl. All rights reserved.
//
import UIKit
class ZonesViewController: UITableViewController {
   var mapView: MapViewController?
   let allZones: [SARZone] = Array(SARZone.allZonesInRealm())
@IBOutlet var tableview: UITableView!
```

La classe di riferimento per il menu delle Zone è la ZonesViewController.swift, di tipo UITableViewController, ed ha i seguenti attributi:

- mapView, di tipo MapViewController, che non è altro che la mappa principale dell'app, sulla quale ci si sposta con la scelta della zona;
- allZones, che è un array, contenente tutte le zone tutte le zone disponibili, ottenute interrogando il database locale;
- tableview, che è il menu di selezione delle zone. La scritta
  @IBOutlet sta a significare che la variabile in questione è un
  collegamento diretto ad un elemento presente nel
  Main.storyboard, ovvero la sezione dell'ambiente di sviluppo
  dove sono collocati gli elementi grafici ed i vari collegamenti ad
  essi.

### STRUTTURA DELLA CLASSE ZONESVIEWCONTROLLER

La classe zonesViewController è formata dai seguenti metodi:

```
override func numberOfSectionsInTableView(tableView: UITableView) -> Int {
   // #warning Incomplete implementation, return the number of sections
   return 1
}
```

Il metodo numberOfSectionsInTableView è uno dei molti metodi nativi, ovvero che va ridefinito in base agli usi che lo sviluppatore ne farà; questo metodo in particolare ritorna il numero di sezioni che il menu avrà. Di default, tutte le tableView ne hanno solo una, ed è così anche nel nostro caso.

```
override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    // #warning Incomplete implementation, return the number of rows
    return allZones.count
}
```

Il metodo tableView viene ridefinito molte volte: in questo caso, gli viene passato l'elemento "tableView", simboleggiante la tabella contenente le varie zone, ed ha il compito di definire il numero di righe che il menu avrà; per far ciò basta contare il numero di elementi che ha il vettore allZones.

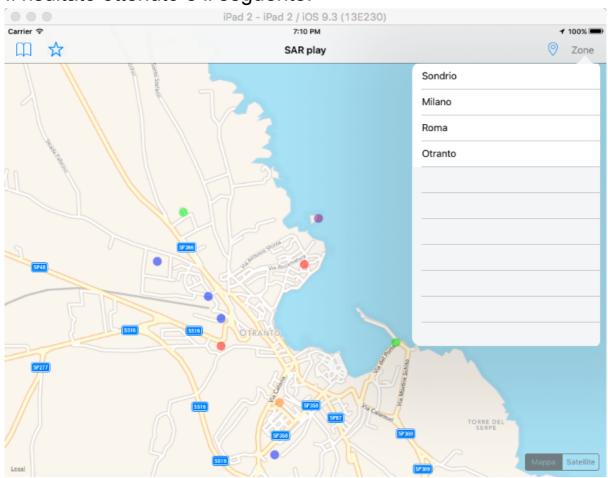
```
override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell {
    let cell:UITableViewCell = self.tableView.dequeueReusableCellWithIdentifier(cellReuseIdentifier) as UITableViewCell!
    cell.textLabel?.text = allZones[indexPath.row].title as String
    return cell
```

Anche in questo caso, al metodo tableView viene passata la stessa tableView, ma la funzione del metodo è totalmente diversa: il metodo viene richiamato tante volte quante sono le celle (vedi metodo precedente), e per ciascuna di esse viene modificata la propria label col nome di una zona.

```
override func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath: NSIndexPath) {
    mapView?.currentZone = allZones[indexPath.row]
    dismissViewControllerAnimated(true, completion: nil)// chiudi la finestrella
```

L'ultimo metodo di questa classe viene invocato ogni volta che si verifica una selezione di un elemento del nostro menu: nel nostro caso si vuole che ci si sposti su una determinata zona, pertanto si va a modificare l'attributo "currentZone" (zona corrente in cui ci si trova) dell'oggetto mapView con l'elemento da noi scelto, cercandolo all'interno del vettore allZones. Una volta selezionato, la schermata del menu viene chiusa, in modo che l'utente possa usare la mappa per intero e senza menu aperti.

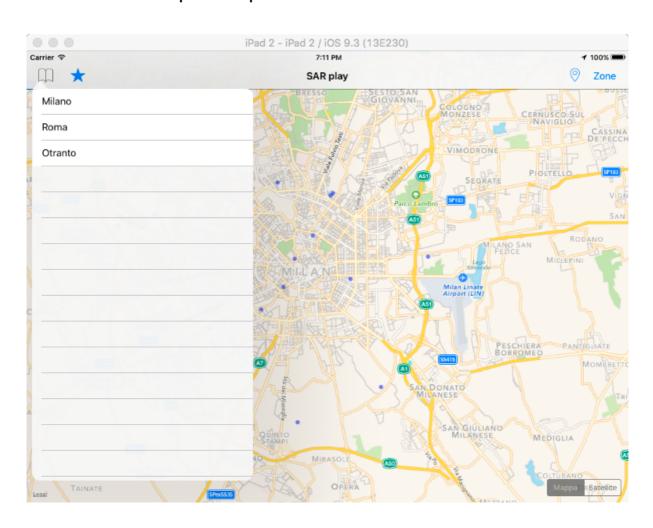
### Il risultato ottenuto è il seguente:



### IL MENU DEI PREFERITI

Per una maggiore comodità e semplicità di utilizzo, è stata pensata una zona preferiti, in maniera che l'utente possa salvare le zone che visita maggiormente senza il bisogno di doverle cercare ogni volta tra tutte le zone disponibili. Il funzionamento è molto simile al menu dei preferiti, infatti il menu dei preferiti è formato dagli stessi elementi del più precedente, però νi è il menu in pulsante aggiungere/rimuovere la zona dei preferiti, rappresentato da una stella piena, quando la zona è tra i preferiti, e una stella vuota quando la zona non è tra i preferiti.

### Il risultato sarà questo qua:



### INSERIRE UNA ZONA TRA I PREFERITI

```
@IBAction func touchDown(sender: AnyObject) {
    if currentZone?.isBookmarked == false{
        currentZone?.bookmark(true)
        setBookmark.image = UIImage(named: "bookmark.png")
    }
    else{
        currentZone?.bookmark(false)
        setBookmark.image = UIImage(named: "not-bookmark.png")
    }
}
```

Il frammento di codice riportato qua sopra viene richiamato quando viene premuto il pulsante di inserimento/rimozione della zona corrente dai preferiti.

Ogni oggetto di tipo SARZone ha come attributo un valore "bookmarked\_" di denominato boolean il tipo metodo .isBookmarked(), che ritorna "TRUE" se la zona è tra i preferiti, e ritorna false in caso contrario. Qualora la zona non sia tra i preferiti, l'immagine del bottone "aggiungi/rimuovi" è una stella vuota: in caso di pressione, bisogna innanzitutto aggiungere la zona ai preferiti, utilizzando il metodo .bookmark(true), e in seguito l'immagine, scegliendo la stella piena.

Nel caso che la zona sia già tra i preferiti, bisogna fare il procedimento opposto, ovvero rimuovere la zona corrente dai preferiti, usando il metodo .bookmark(false), ed impostare l'immagine del bottone con la Stella vuota

### STRUTTURA DELLA CLASSE BOOKMARKSVIEWCONTROLLER

```
//
// BookmarksViewController.swift
// SARplay
//
// Created by Michele Longhi on 24/02/16.
// Copyright © 2016 Visuality srl. All rights reserved.
//
import UIKit
class BookmarksViewController: UITableViewController {
   var zones: [SARZone] = Array(SARZone.allBookmarkedZones())
   var mapView: MapViewController?
```

Anche in questo caso, la classe bookmarksViewController è di tipo UITableViewController, dato che nell'app è presente un menu conenente tutte le zone preferite. Gli attributi di questa classe, sono molto simili a quelli della ZonesViewController, infatti gli attributi sono:

- zones, ovvero un vettore contenente tutte le zone che hanno il valore "bookmarked\_" impostato come "true";
- mapview, di tipo MapViewController, ovvero la variabile che si riferisce alla mappa della schermata principale

La classe bookmarksViewController è inoltre formata dai seguenti metodi:

```
override func numberOfSectionsInTableView(tableView: UITableView) -> Int {
    // #warning Incomplete implementation, return the number of sections
    return 1
}
```

Come nel caso precedente, è il metodo che definisce il numero di sezioni di una tableView, e, come nel caso precedente, ritorna 1.

```
override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
   // #warning Incomplete implementation, return the number of rows
   return zones.count
}
```

Anche in questo caso, il metodo riportato sopra ha la funzione di definire il numero di righe che il menu dovrà contenere; per far ciò, è sufficiente contare tutti gli elementi contenuti del vettore "zones", poiché ciascun elemento di esso occuperà una cella del menu.

```
override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell {
    let cell:UITableViewCell = self.tableView.dequeueReusableCellWithIdentifier(cellReuseIdentifier) as UITableViewCell!
    // Configure the cell...
    cell.textLabel?.text = zones[indexPath.row].title as String
    return cell
```

Questo metodo ridefinito ha il compito di dare un nome a ciascuna cella del menu, andando a prelevarlo dal nome di ciascuna zona contenuto nel vettore "zones"; come nel caso precedente, questo metodo viene invocato ad ogni apertura del menu a tendina tante volte quante sono le zone contenute nell'array "zones".

```
override func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath: NSIndexPath) {
    mapView?.currentZone = zones[indexPath.row]
    dismissViewControllerAnimated(true, completion: nil)// chiudi la finestrella
}
```

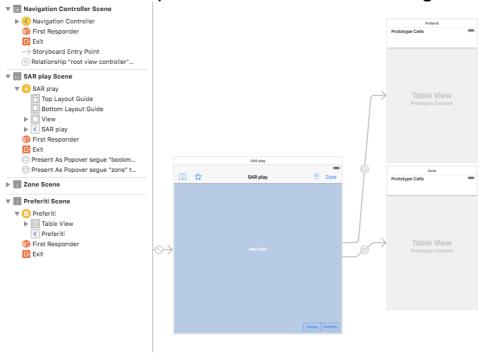
La funzione in questione viene richiamata ogni qualvolta viene registrato un evento di pressione su una qualunque cella del menu: nel nostro caso, ci si vuole spostare sulla zona da noi scelta, pertanto si imposta il valore "currentZone" della mappa, contenuta nella schermata principale, con il valore dell'array nella posizione da noi scelta.

Tuttavia, sia nel caso precedente, sia in questo, una volta premuta la cella contenente la zona desiderata non succederebbe nulla, se non ci fossero i "segue": più avanti verrà spiegata la loro funzione.

### **I SEGUE**

Un Segue è l'operazione che ti permette di spostarti da un ViewController A ad uno B che si trova in una posizione successiva rispetto alla partenza. Oltre a spostarsi semplicemente, i segue permettono anche di trasferire dati da una view ad un'altra.

Nel nostro caso i segue sono due, e collegano i bottoni "Zone" e "Preferiti" alle rispettive tableView, come in figura:



I due segue hanno un nome identificativo, in maniera che siano univoci e che non vi possano essere ambiguità qualora essi vengano richiamati.

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
   if let newController = segue.destinationViewController as? ZonesViewController {
      newController.mapView = self
      TypeLocation.image = UIImage(named:"automatico")
   }

if let newController = segue.destinationViewController as? BookmarksViewController {
      newController.mapView = self
      TypeLocation.image = UIImage(named:"automatico")
}
```

Il frammento di codice sopra riportato viene richiamato ogni qualvolta si cerca di modificare l'attributo "currentZone", poiché esso si trova in una view diversa da quella corrente.

### CONCLUSIONI

Lo scopo della collaborazione ITIS-Visuality srl era sin da subito la realizzazione di un'app funzionante da pubblicare sullo store, in quanto in caso contrario sarebbe stata tutta una grande perdita di tempo. Oltre a far vedere a noi alunni cosa significhi veramente lavorare in gruppo, quest'esperienza ci ha dato le basi per lavorare su una piattaforma a noi sconosciuta. Lavorare in gruppo, con a capo un developer per professione e con molti anni di esperienza è stato certamente positivo, e ci ha dato un primo approccio ad un eventuale futuro lavorativo nel mondo dello sviluppo di applicazioni.

Personalmente, la ritengo molto positiva come esperienza, in quanto possessore di prodotti Apple da parecchio tempo, l'idea di vedere e partecipare alla realizzazione di un applicativo per tali piattaforme mi ha sin da subito incuriosito; naturalmente, essendo un progetto non facente parte del programma, indubbiamente avrebbe meritato più spiegazioni sulle metodologie di lavoro. Ciononostante, a breve l'applicazione sarà ultimata, e si spera sempre a breve che finisca sullo store di Apple.

### **BIBLIOGRAFIA**

Per la realizzazione dell'applicazione SARPlay sono stati consultati i seguenti siti web:

- <a href="http://stackoverflow.com/">http://stackoverflow.com/</a>→ sito internet dedicato a programmatori di tutto il mondo, nel quale si espongono i propri dubbi o malfunzionamenti e si viene aiutati; essendo una community di programmatori di tutto il mondo, la lingua del sito è l'inglese
- <a href="https://www.xcoding.it/">https://www.xcoding.it/</a> → sito web dedicato allo sviluppo e alla progettazione di app multipiattaforma, utilizzato in particolare da chi è alle prime armi; sito in italiano
- https://developer.apple.com/library/ios/navigation/ → portale di Apple contenente tutta la documentazione riguardante il mondo dei dispositivi mobili della casa; la lingua è l'inglese
- https://swift.org/documentation/
  interamente al suo linguaggio di programmazione, in lingua inglese