



Bentornati a tutti, al nostro appuntamento settimanale con le guide a DashCode. Oggi studieremo come trasformare una WebApp in un'App Nativa. Sottolineo poi come questa che iniziamo oggi sia una guida suddivisa in più capitoli. Già la parte di oggi è particolarmente lunga e complessa.

Requisiti:

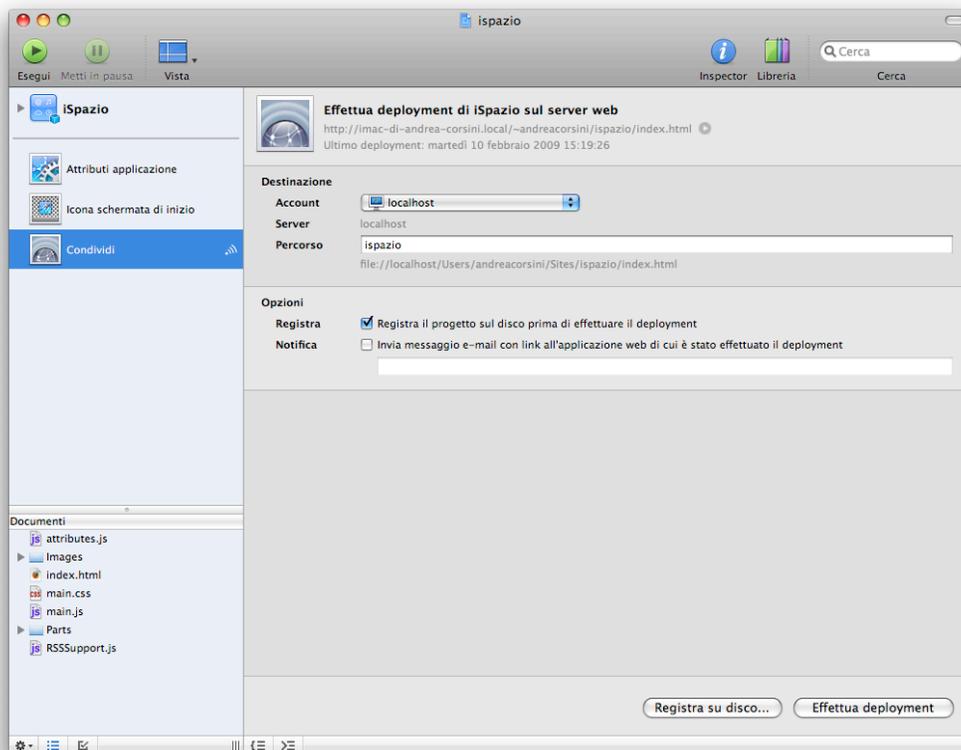
- Developer Tools 3.0 (Xcode, DashCode, iPhone Simulator)

Obiettivi:

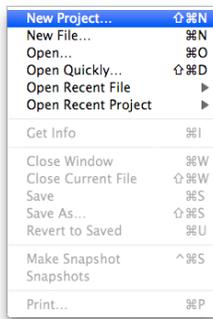
- Approfondire la conoscenza di DashCode e Xcode
- Assaggiare che è Interface Builder
- Preparare una WebApp per diventare un'App Nativa
- Condividere file tra DashCode e Xcode

Procedimento:

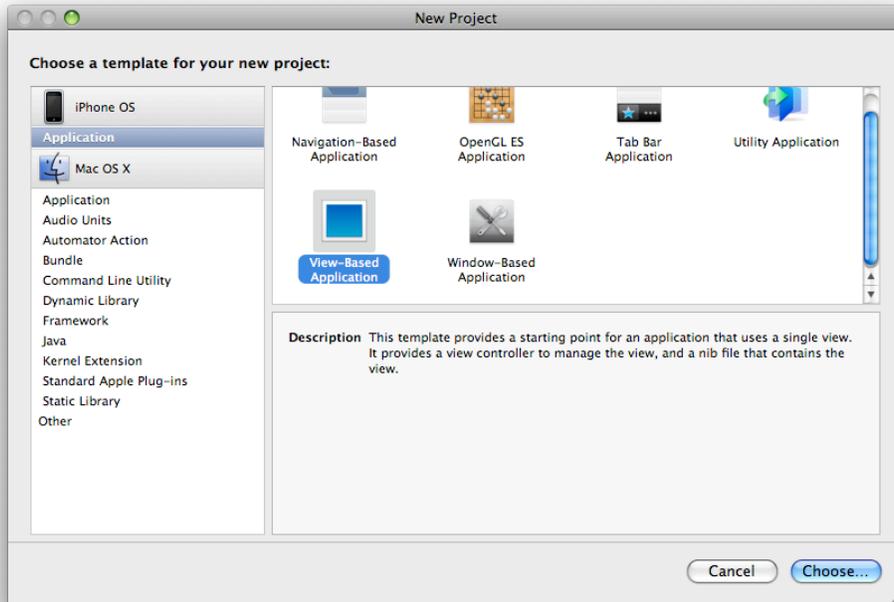
1. Apriamo **DashCode**, e apriamo il nostro progetto creato precedentemente. Spostiamoci nella sezione **Condividi** e modifichiamo le impostazioni a nostro piacimento. Poi premiamo su **Effettua deployment**:



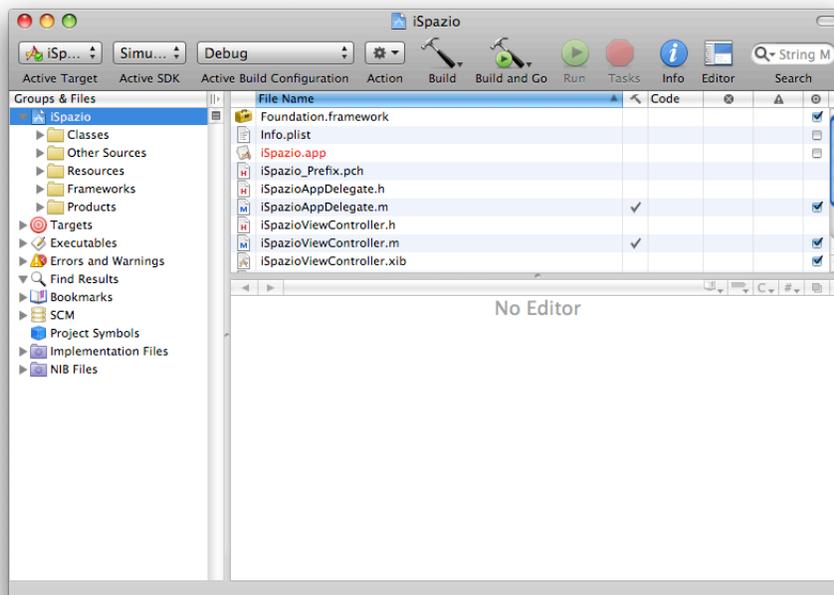
2. Ora chiudiamo **DashCode** e apriamo **Xcode**, dalla barra degli strumenti scegliamo di creare un **New Project**.



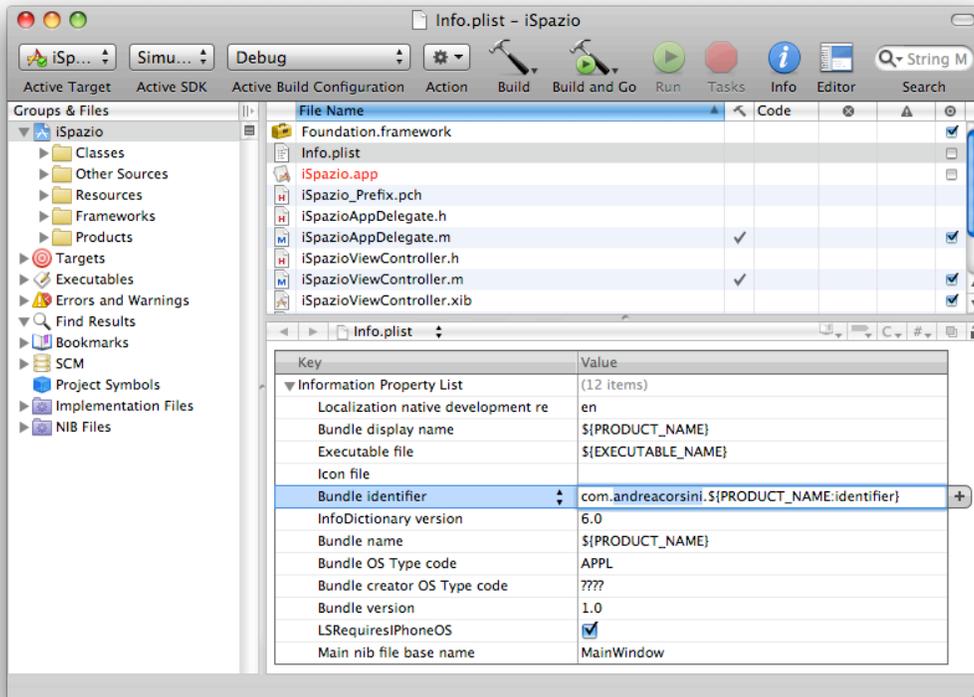
3. Scegliamo a sinistra una **Application** per **iPhone OS** ed a destra **View Based**.



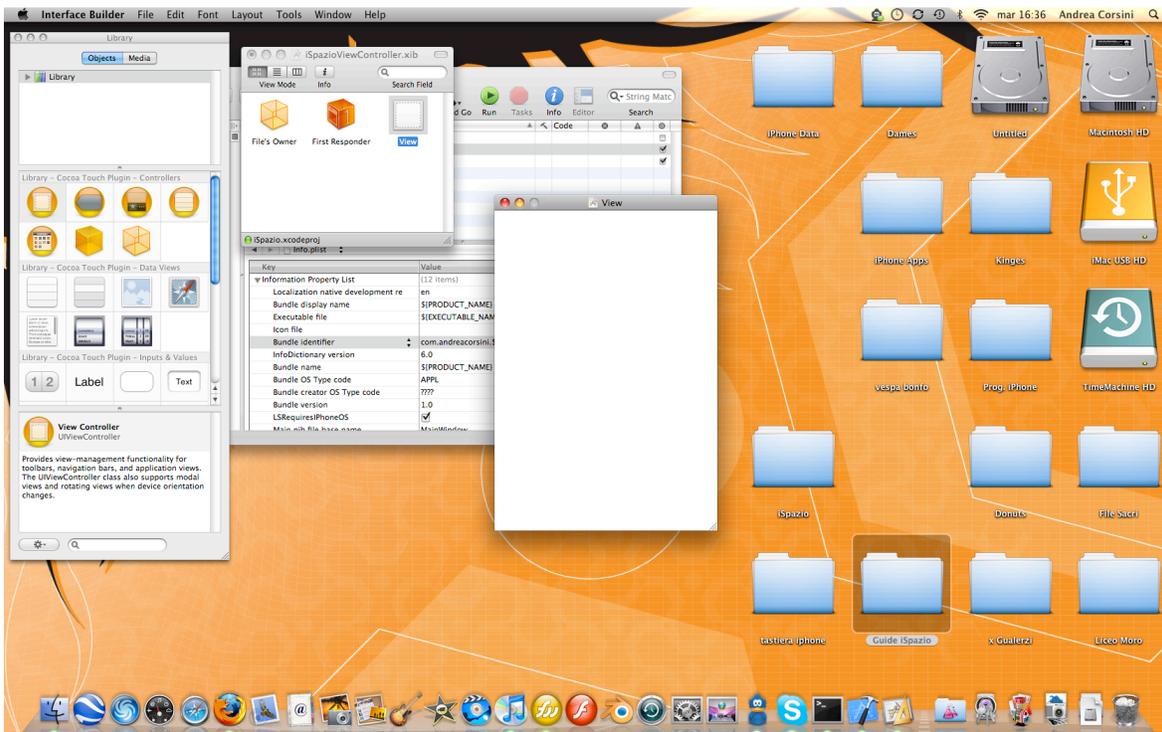
4. Ci troveremo davanti a qualcosa di simili a qui sotto. Dapprima dobbiamo ricordarci che **Xcode**, a differenza di **DashCode**, non è tradotto in italiano ed è quindi completamente in inglese. Ora facciamo una panoramica generale. A sinistra troviamo tutte le parti che compongono l'App non ancora compilata, suddivisi in categorie. In alto troviamo vari pulsanti ed impostazioni quali per esempio **Active SDK** e **Active Build Configuration** che ci saranno utili prossimamente e i due pulsanti **Build** e **Build and Go**. Poi nella metà destra alta abbiamo un elenco dei file presenti, ma non organizzati come a sinistra. Nella metà destra bassa abbiamo un editor di testi che all'evenienza si può trasformare anche in una sorta di **Anteprima** rudimentali.



5. Per prima cosa clicchiamo una sola volta sul file **Info.plist** e vedremo che l'editor lo aprirà. Adattiamo i parametri alle nostre esigenze.

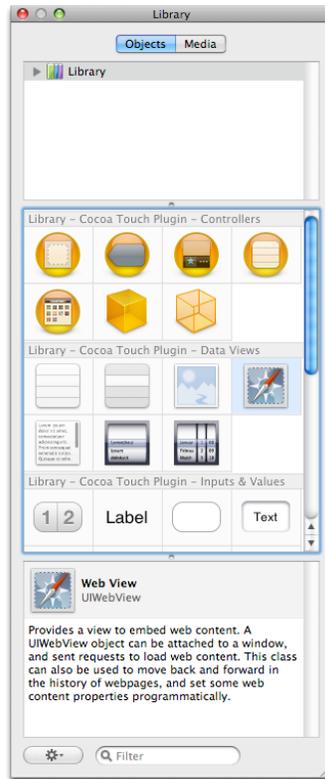


6. Dalla colonna di sinistra scegliamo **Resources** e poi facciamo doppio click su **iSpazioViewController.xib**. Si aprirà automaticamente **Interface Builder**.

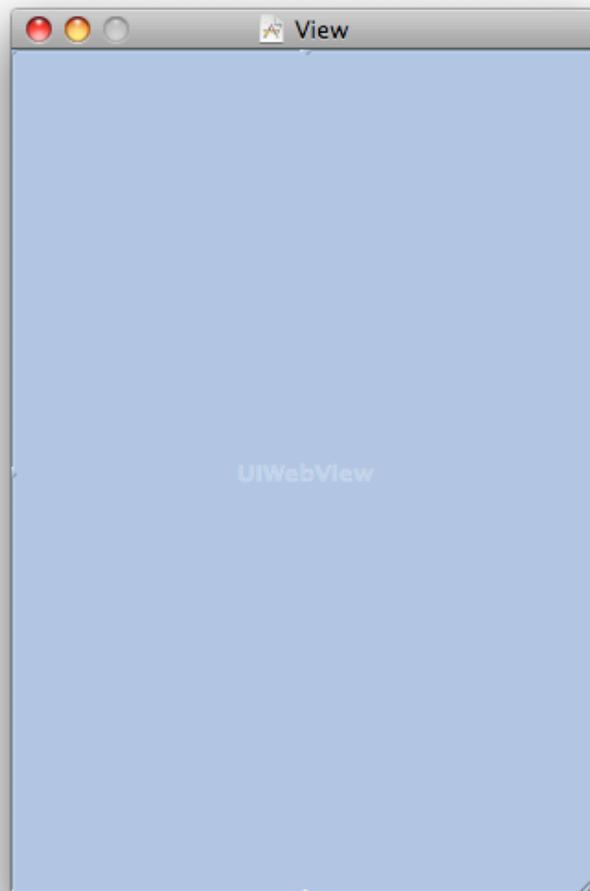


7. Sulla sinistra notiamo una finestra chiamata **Library**. Da questa possiamo scegliere tutti i componenti grafici, e non, che possiamo comodamente trascinare all'interno dell'altra finestra (adesso completamente bianca) che rappresenta, in scala reale, quello che si vedrà sull'iPhone una volta lanciata l'applicazione. Noi sce-

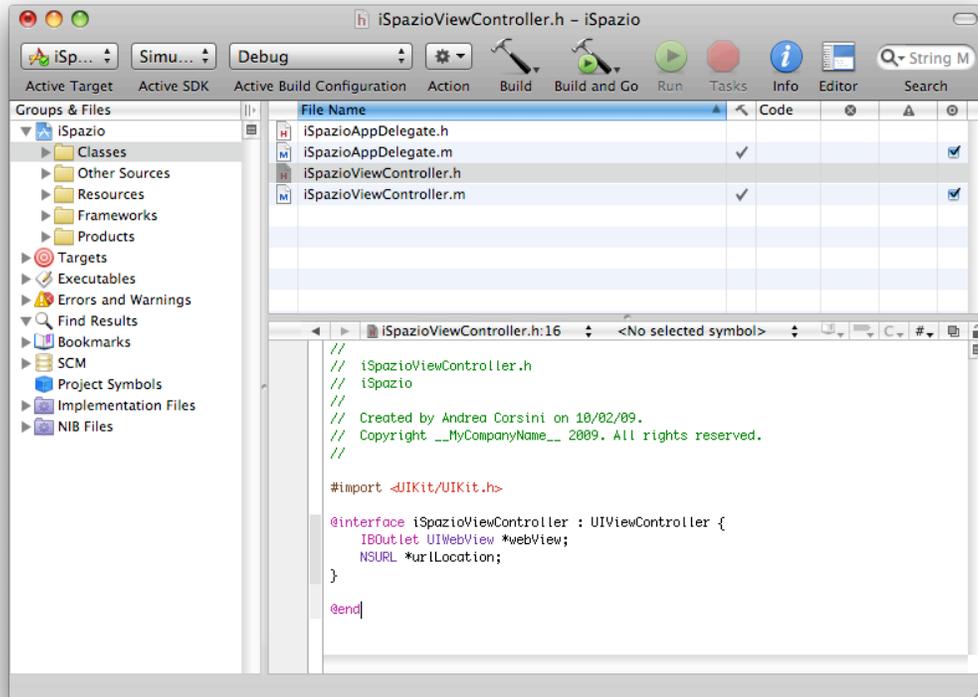
gliamo **Web View** dalla categoria **Library - Cocoa Touch Plugins - Data Views** e la trasciniamo nella finestra bianca. Il nostro componente si ridimensionerà automaticamente, tanto da occupare tutta la finestra.



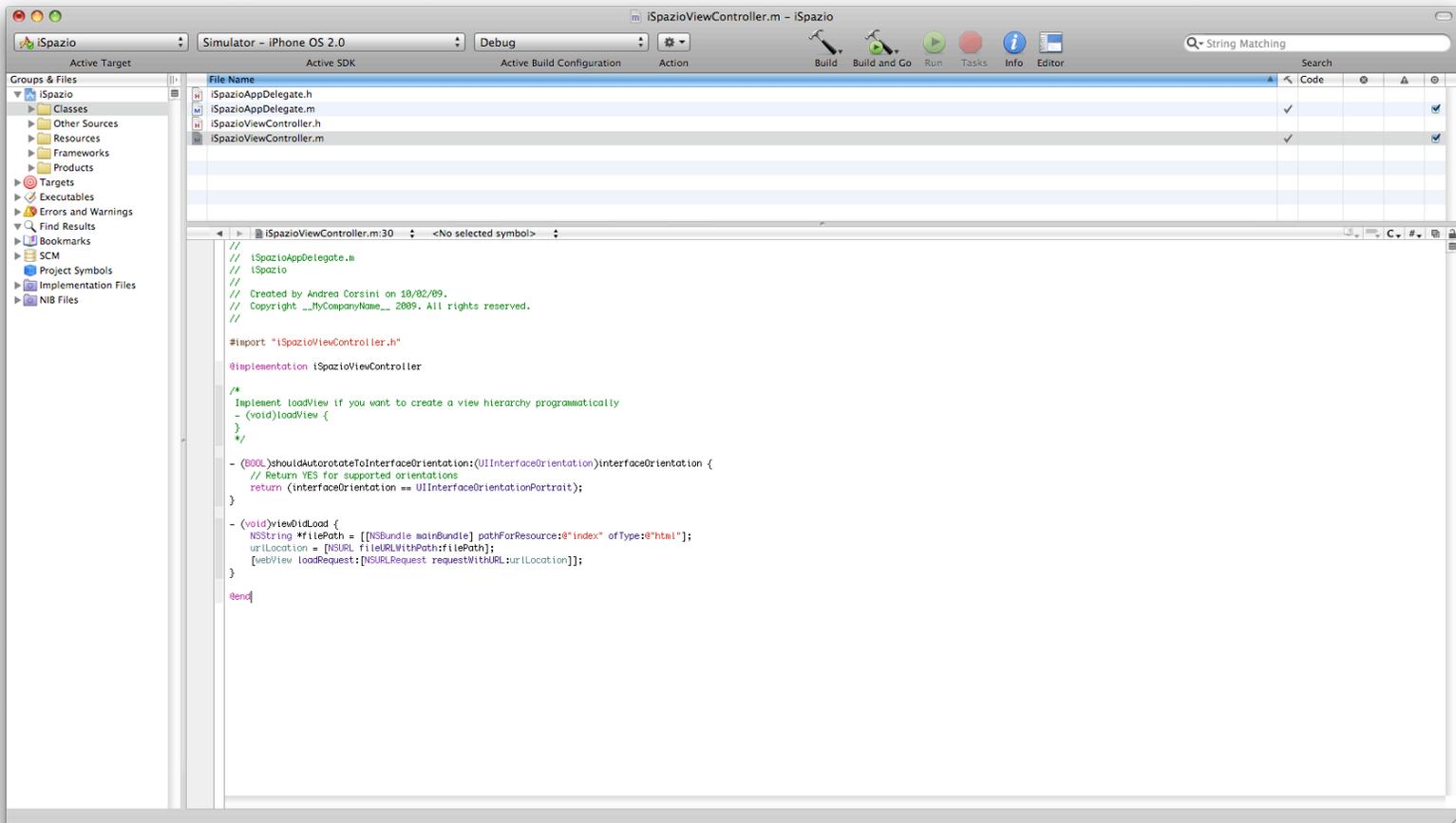
8. Otterremo qualcosa come qui sotto. Ora salviamo il file grafico di progetto.



9. Torniamo a **Xcode**. Dobbiamo ora modificare alcuni file, che troviamo nel gruppo **Classes** sulla sinistra, affinché la nostra App funzioni correttamente. Per primo modifichiamo il file **iSpazioViewController.h**:



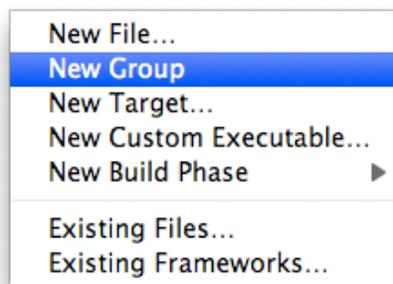
10. Salviamo il file e passiamo a modificare **iSpazioViewController.m**:



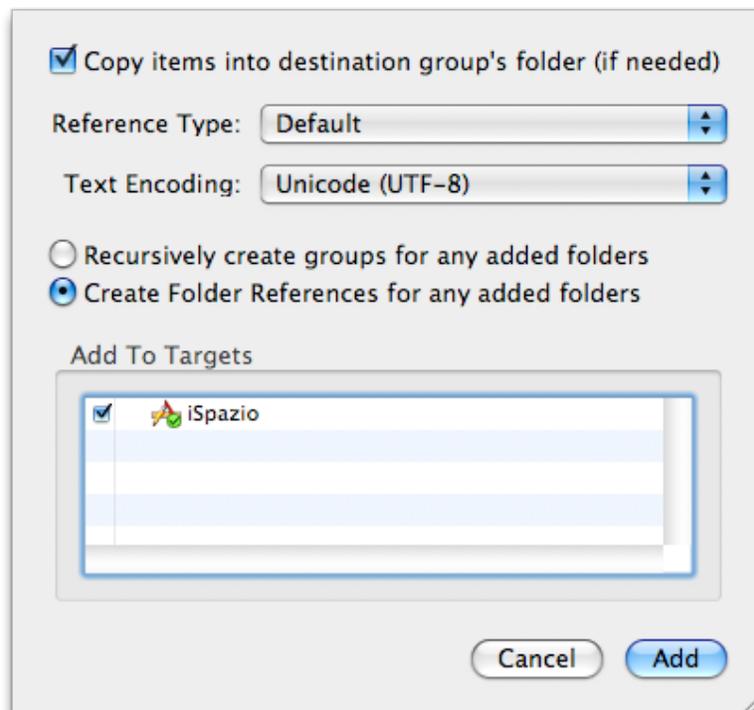
11. Completato questo processo e salvati i file possiamo tornare ad **Interface Builder**. Quando ci si è aperto per la prima volta questa applicazione abbiamo parlato di due finestre, ma ne erano presenti tre. La terza contiene la parte più progettuale e sistemica di questo progetto in versione grafica. Premiamo con il tasto destro sull'icona chiamata **File's Owner**, ci si aprirà un attraente pannello nero. Premiamo il cerchietto bianco di fianco alla scritta **webView** e trasciniamolo fino alla nostra finestra dove avevamo prima posizionata la **Web View**.



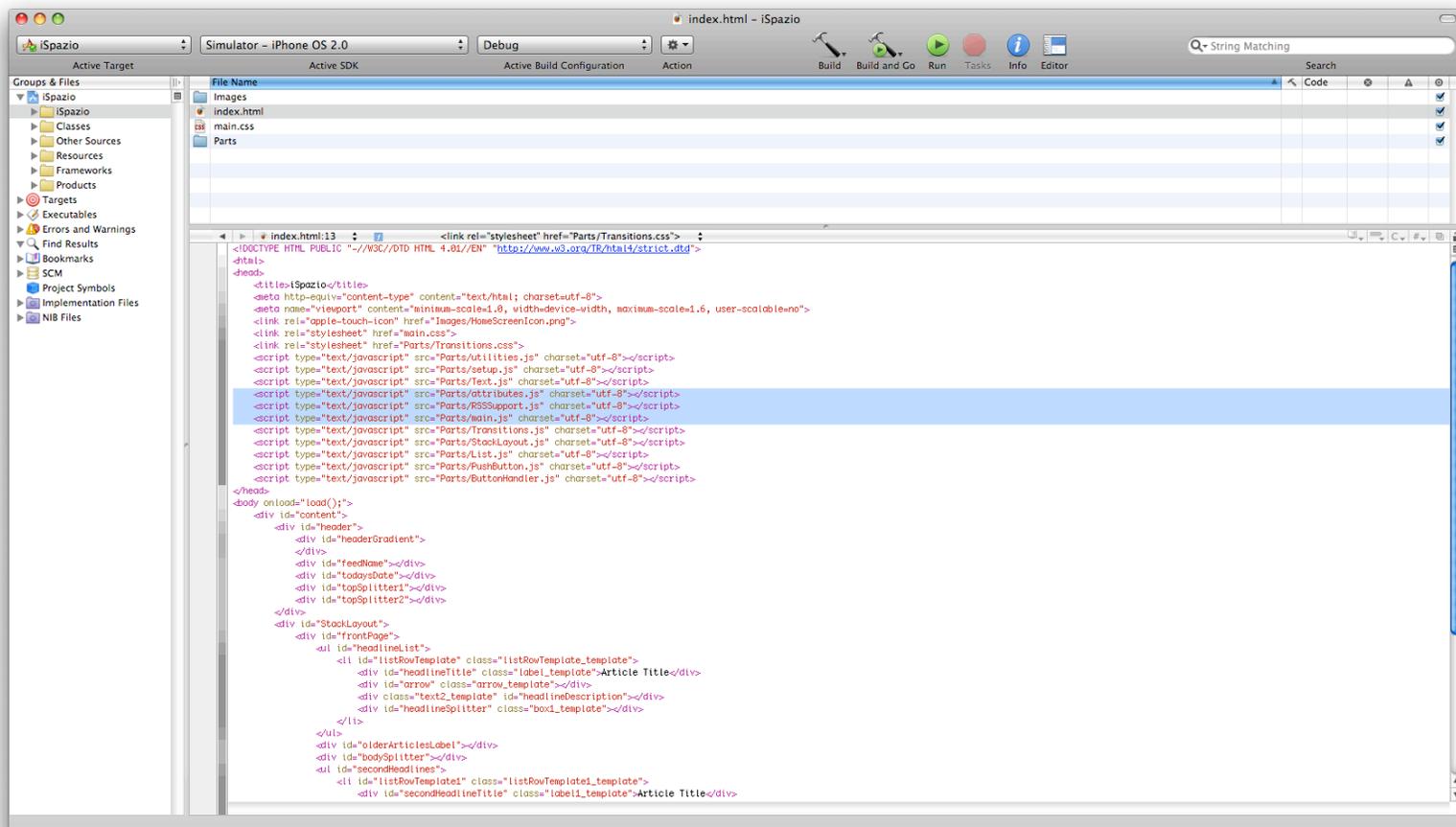
12. Salviamo tutto e chiudiamo pure **Interface Builder**. Rieccoci ad **Xcode**. Dalla colonna di sinistra premiamo con il tasto destro e scegliamo **Add -> New Group**. Chiamiamolo **iSpazio**.



13. Con il **Finder** andiamo a trovare quella cartella in cui è stata salvata la nostra WebApp con **DashCode**. Il percorso dovrebbe essere questo: "*Macintosh HD/~/NomeUtente~/Siti/ispazio*". Una volta entrati in questa cartella spostiamo i tre file con estensione *.js (**attributes.js**, **main.js**, **RSSsupport.js**) nella cartella **Parts**. Poi trasciniamo tutti i file presenti nella cartella **ispazio**, in cui ci eravamo portati prima, all'interno del gruppo **iSpazio**, che abbiamo precedentemente creato in **Xcode**. Appena fatto questo ci si aprirà un nuovo pannello che ci chiederà di decidere i parametri per la copia dei file. Impostiamo come qui sotto:



14. Ora sempre dalla colonna di sinistra, nel gruppo **iSpazio**, scegliamo **index.html**. Per questo file dobbiamo effettuare delle modifiche alle prime righe di codice.



15. Una volta modificato il file e salvato, abbiamo finito. Dalla barra degli strumenti controlliamo che **Active SDK** sia impostato su **Simulator - iPhone OS 2.0** e che **Active Build Configuration** sia su **Debug**. Ora premiamo il tasto **Build and Go**. Per qualche secondo **Xcode** lavorerà e compilerà la nostra app, poi si aprirà **iPhone Simulator** e si lancerà automaticamente l'applicazione.



Questa era l'ultima fase del progetto di oggi. Abbiamo ottenuto un'App Nativa che mostra una breve anteprima di tutti i contenuti che vengono pubblicati su iSpazio.

Se aveste problemi, potete scaricare il file di progetto

Nella prossima puntata personalizzeremo ancora di più quest'app e vedremo come trasferirla e farla funzionare sul nostro iPhone o iPod Touch anche senza una licenza di developer Apple. Con questo concludo e vi saluto. Grazie a tutti dell'attenzione,

Andrea Corsini per iSpazio.net