

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Facoltà di Ingegneria “Enzo Ferrari” – Sede di Modena

Corso di Laurea Specialistica in Ingegneria Informatica

Progettazione e Sviluppo dell’Applicazione Web Share Mobile per Dispositivi Mobili Multiplatforma

Relatore:

Chiar.mo Prof. Sonia Bergamaschi

Correlatore:

Prof. Riccardo Martoglia

Correlatore Esterno:

Ing. Andrea Prandini

Elaborato di Laurea di:

Marco Stanzani

Situazione Iniziale

Presenza di due Web Application:

- Web Share → distribuzione e catalogazione di documenti
- PUMA → gestione utenti
 - gestione autenticazione
 - gestione autorizzazioni utente

Obiettivo

Creazione dell'APP di Web Share per dispositivi **Apple** e **Android**

- Download dei documenti
- Visualizzazione dei documenti

I Dispositivi Mobili

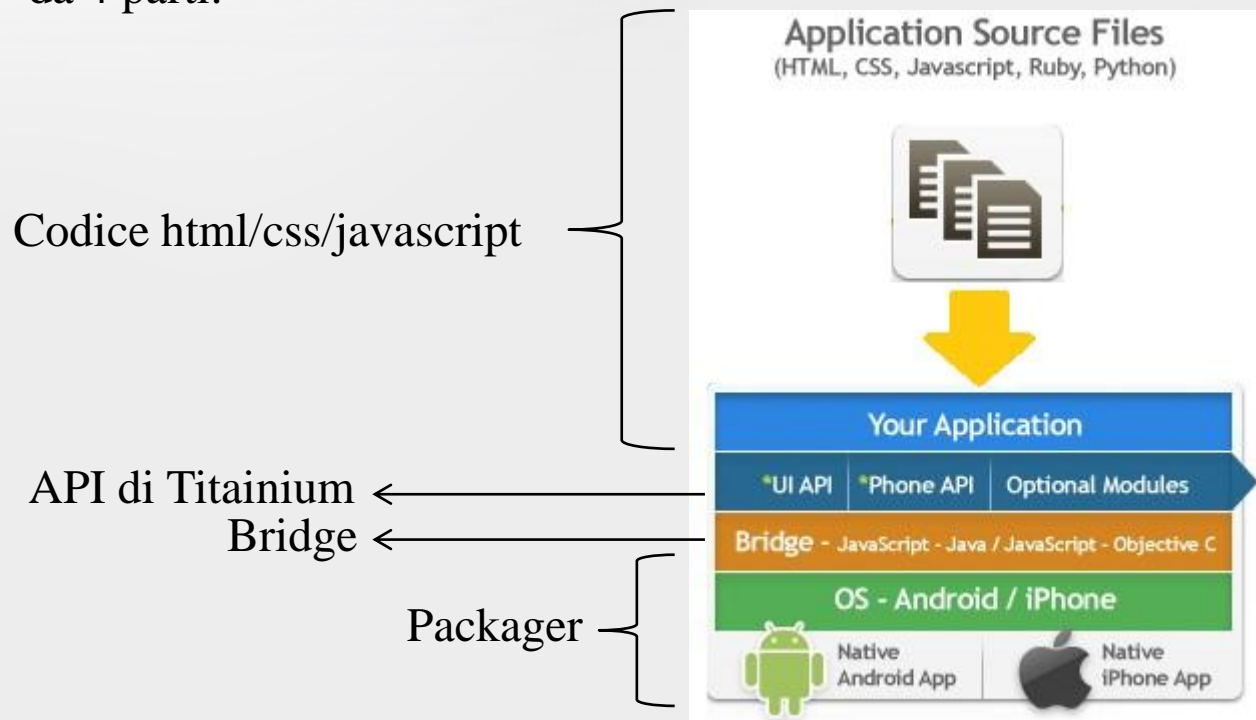


- Comunicazione tra utente e dispositivo limitata
- Potenzialità HW ridotte rispetto i PC
- Grande eterogeneità hardware e software fra i vari dispositivi mobili

Appcelerator Titanium

Framework permette di creare applicazioni native sia per dispositivi mobili che per applicazioni desktop, partendo da un codice comune scritto in Java Script.

La struttura App è composta da 4 parti:



Bridge - Packager

In particolare, il bridge e il packager, sfruttano tre funzionalità principali:

1. Pre-compiler: il suo ruolo è quello di prendere il codice JavaScript dell'applicazione e ottimizzarlo e di creare una gerarchia di dipendenza di tutte le API di Titanium utilizzate dall'applicazione,
2. Front-end compiler: Il suo compito è quello di generare il codice specifico per la piattaforma nativa,
3. Platform compiler & packager: Funzionalità che crea il pacchetto da somministrare ai enti di giudizio gestiti da Apple e Google.



Web Share Mobile

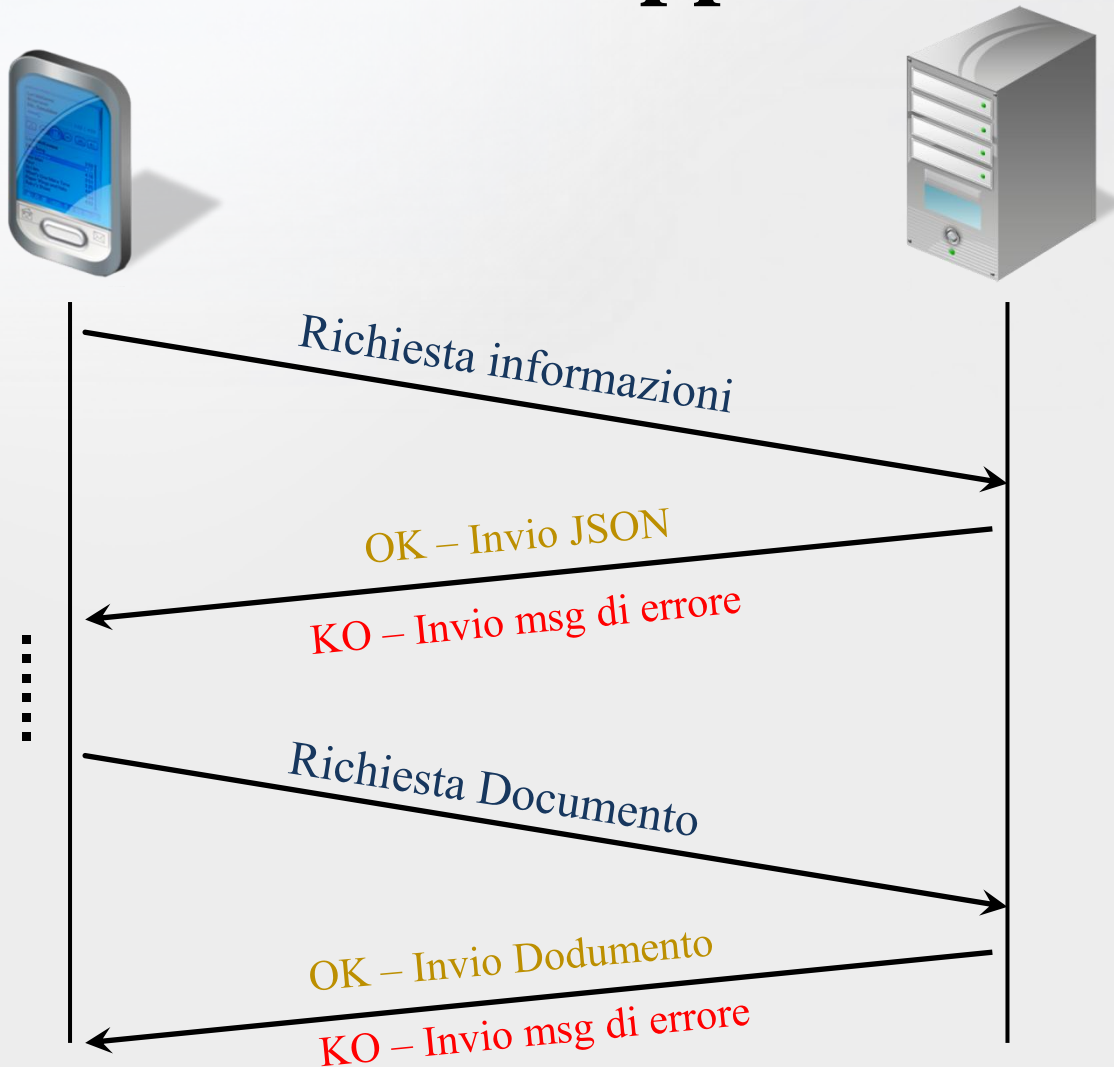
Client:

- Scaricare le informazioni dei documenti più recenti
- Scaricare e visualizzare i documenti di interesse
- Inviare i documenti come allegati di una email
- Creare una pagina delle preferenze esterna all'App
- Cancellazione automatica delle informazioni e dei rispettivi file (se esistono)

Server:

- Filtro che gestisce l'autenticazione dell'utente
- Creazione e invio di un oggetto JSON, che contiene le informazioni dei documenti
- Invio del file richiesto

Funzionamento dell'App

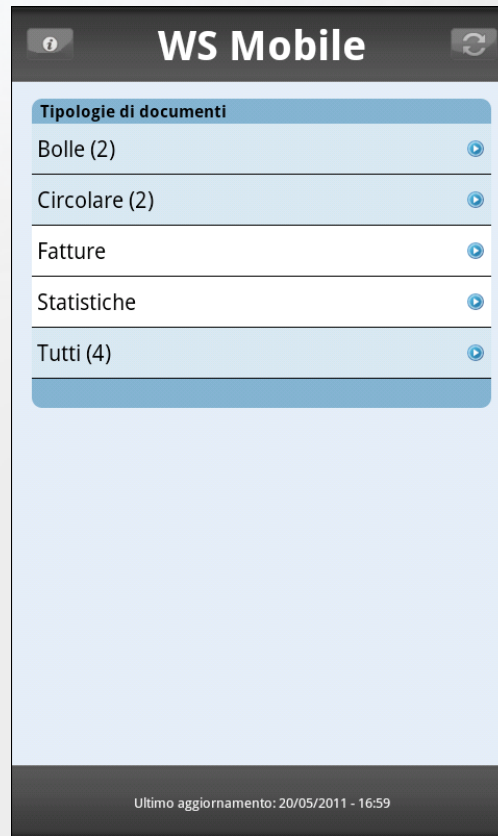


Struttura dell'APP – Finestra Archive List

iOS



Android



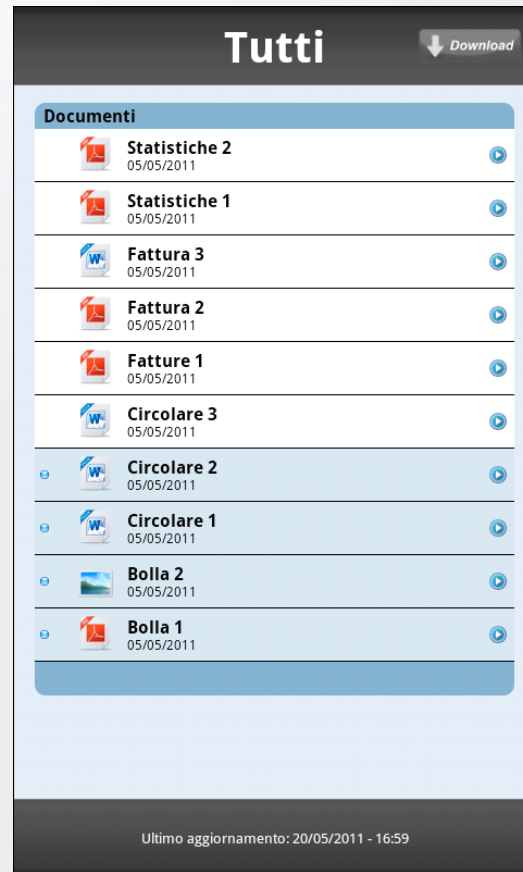
- Visualizza la lista dei archivi alla quale appartiene almeno un documento
- Permette il download delle informazioni dei documenti

Struttura dell'APP – Finestra Document List

iOS



Android



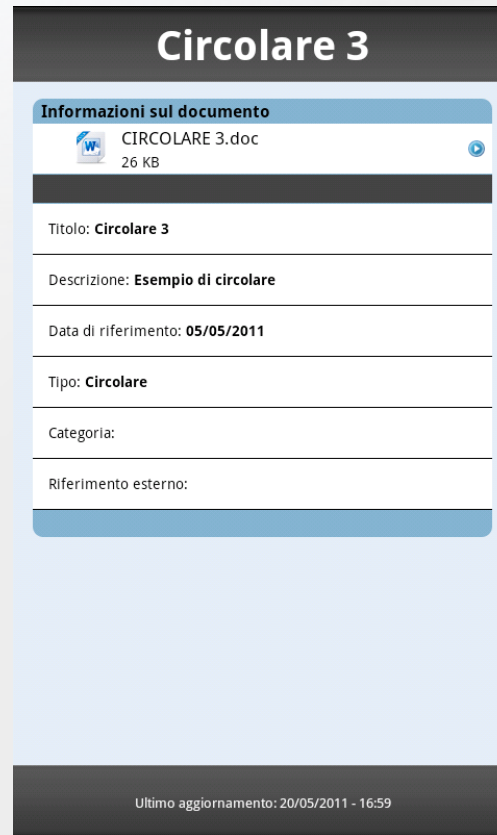
- Mostra tutti i documenti oppure tutti quelli appartenenti a un archivio
- Permette il download di tutti i documenti visualizzati all'interno della lista

Struttura dell'APP – Finestra Document Detail

iOS



Android



- Mostra tutte le informazioni riguardanti a un documento
- Permette il download del documento scaricato

Struttura dell'APP – Pagina delle Preferenze

iOS



Android



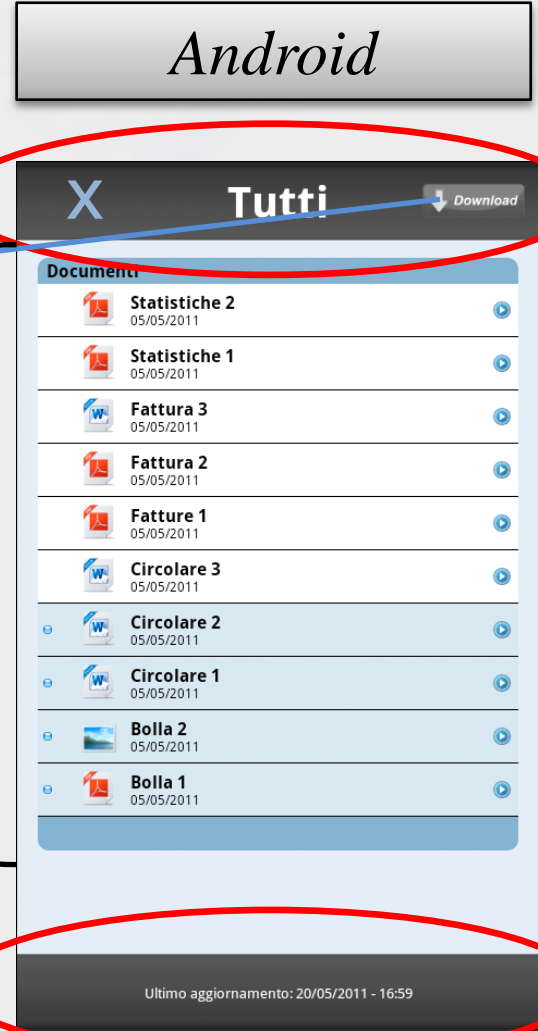
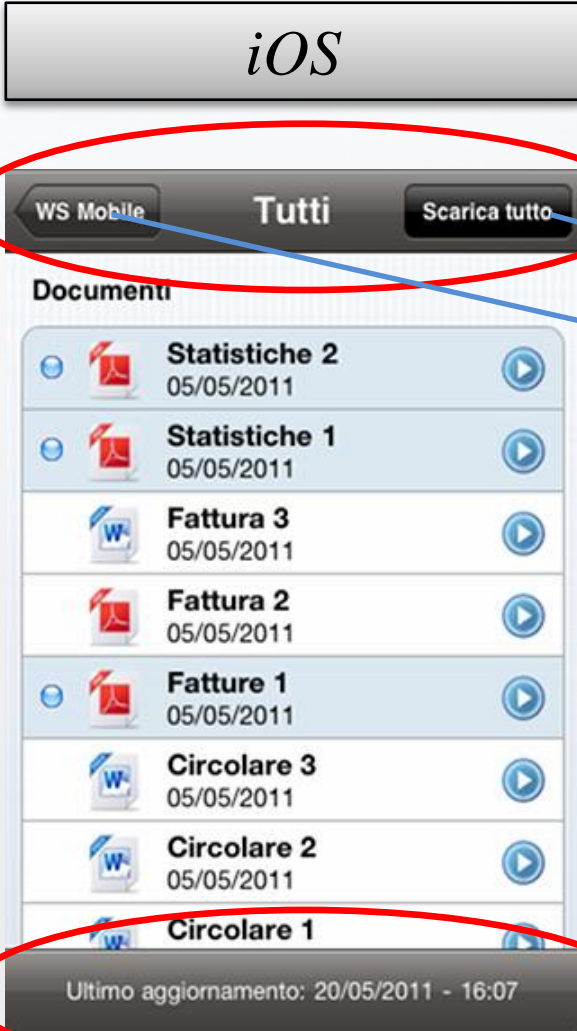
All'interno di questa pagina vengono inseriti:

- Nome utente
- Password
- URL del Server
- Numero di documenti
- Se si desidera salvare la password

Componenti Grafici

iOS

Android



Navigation Bar

Pulsante

Back Button

Tabella

Tool Bar

Problematiche nello sviluppo della GUI con Android

- Non esistono *Navigation Bar* o *Tool Bar*
- I pulsanti non hanno una bella grafica e si è dovuto creare un'immagine da inserire nei bottoni
- Difficoltà nel creare una tabella con dimensioni variabile
- Mancanza di diversi effetti grafici rispetto dispositivi con iOS
- Ci sono sei tipi di risoluzioni grafiche diverse;

Visualizzazione di un Documento - iOS

Visualizzazione di qualsiasi tipo di file mediante il componente **Web View**.

```
var wv = Ti.UI.createWebView({
    url: win.file.nativePath
});
win.add(wv);
```

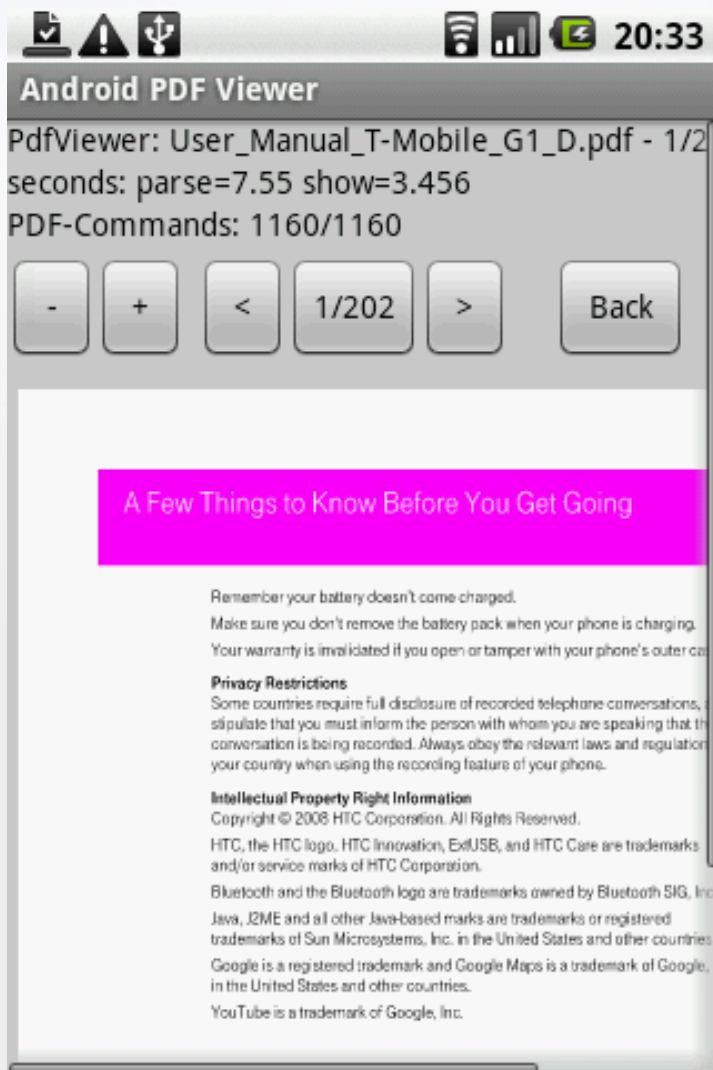
Permette di visualizzare anche:

- Pagine HTML locali o remote
- Vere e proprie Web App

Indicando solamente l'URL della pagina



Visualizzazione di un Documento - Android



Sviluppato un modulo Java che permette l'apertura delle varie tipologie di documenti tramite un'App esterna.

Problematica:

Modulo inserito all'interno del codice sorgente del framework

→ personalizzazione del SDK di Titanium

→ ricompilazione del SDK di Titanium a ogni suo aggiornamento

Soluzione :

A Ottobre Appcelerator a risolto questo problema con l'aggiunta di alcuni moduli che permettono di sviluppare il modulo all'interno del codice dell'App.

Sviluppi Futuri

Anche se la mia applicazione ha risposto a tutti i requisiti richiesti può essere ulteriormente migliorata aggiungendo:

- Gestire i documenti di più utenti che utilizzano lo stesso dispositivo
- Implementare una gestione documentale più avanzata permettendo la cancellazione di singoli elementi scelti dall'utente
- Gestire, oltre che i documenti, anche gli archivi
- Creare un modulo che permette di caricare i documenti sul Server
- Scaricare le informazioni sui nuovi documenti in modalità push
- Implementare l'applicativo Web Share Mobile anche per dispositivi Blackberry

Conclusioni

Al termine del periodo di lavoro l'applicativo è scaricabile da:

- *Apple Store*
- *Android Market*

Valutazioni su Appcelerator Titanium

Ritengo che il Framework sia un ottimo prodotto per i seguenti motivi:

- Sviluppo in parallelo su più piattaforme
- Prodotto in rapida crescita
- Ottimo supporto alla programmazione
- Da pochi mesi è possibile sviluppare anche per Blackberry