

Università della Calabria
Dipartimento di Matematica e Informatica
Corso di Laurea in Informatica

UNIVERSITÀ
DELLA CALABRIA



Tesi di Laurea

Un'applicazione cross-platform per i dipendenti Sintegra

RELATORI:

Prof. Donato D'Ambrosio
Prof. William Spataro

CANDIDATO:

Alessandro Scarlato
Matricola 143390

TUTORS AZIENDALI:

Dott.ssa Serena Ardissonne
Dott. Pasquale Zannino

Anno accademico 2015-2016

Alla mia famiglia e a tutte le persone a me care

Indice

1. INTRODUZIONE	5
1.1. CONTESTO E MOTIVAZIONI	5
2. LE APPLICAZIONI CROSS-PLATFORM	6
2.1. COMPARAZIONE NELLO SVILUPPO DI APP NATIVE E APP MULTIPIATTAFORMA. 6	
2.2. SVILUPPO NATIVO	8
2.3. SVILUPPO MULTI-PLATFORM	9
2.4. PRINCIPALI FRAMEWORK PER LO SVILUPPO DI APP CROSS-PLATFORM.....	10
2.5. IL PATTERN MODEL-VIEW-CONTROLLER.....	12
3. TECNOLOGIE UTILIZZATE NELL'APPLICAZIONE.....	14
3.1. IONIC	14
3.2. STRUTTURA DI UN PROGETTO IONIC.....	16
3.3. IONIC VIEW	19
3.4. APACHE CORDOVA	20
3.5. ANGULARJS.....	21
3.6. PHPMYADMIN.....	23
3.7. GITHUB.....	23
4. PROGETTAZIONE DELL'APPLICAZIONE.....	25
4.1. APPLICAZIONE DI RETE	25
4.2. ARCHITETTURA CLIENT-SERVER	25
4.3. SERVER.....	26
4.4. DATABASE.....	27
5. IMPLEMENTAZIONE DELL'APPLICAZIONE	28
5.1. SVILUPPO DELL'APPLICAZIONE	28
5.2. PANORAMICA DELL'APPLICAZIONE.....	30
5.3. PRINCIPALI FUNZIONALITÀ	33
5.4. ESPORTAZIONE TABELLE IN EXCEL E PDF.....	37
6. CONCLUSIONI.....	39
7. RINGRAZIAMENTI.....	40

1. Introduzione

1.1. *Contesto e motivazioni*

La tesi di laurea è sicuramente un passo molto importante per uno studente e proprio per questo motivo ho voluto fare un'attenta valutazione per essere sicuro della scelta, prendendo in considerazione le tecnologie che non avevo visto durante il mio percorso di studi e che si utilizzano nell'attuale universo informatico. Il mondo mobile mi affascina molto, sia perché è un settore in forte sviluppo, ma anche perché è estremamente gratificante vedere quello che si sta realizzando. Lo scenario mobile è molto complesso per la vastissima molteplicità di device presenti sul mercato, questo rende costoso sviluppare per ogni dispositivo ed il relativo sistema operativo. Questa grande varietà di device e sistemi operativi mi ha spinto a studiare e a conoscere delle nuove tecnologie che permettono di sviluppare applicazioni cross-platform, in quanto consentono di scrivere una sola volta il codice e di compilarlo in modo da produrre più app, ognuna destinata ad una piattaforma diversa. La mia scelta è ricaduta su una tesi in azienda, che mi ha permesso di cimentarmi in una nuova esperienza, iniziando a comprendere quelle che sono le realtà lavorative e di come le idee di un progetto si realizzino in un team. Lo sviluppo di questa app mi permette di comprendere quali sono gli step per la realizzazione di una eventuale futura applicazione, così da indirizzarmi meglio sull'ambito lavorativo in cui vorrò cimentarmi.

Lo scopo era quello di realizzare un'applicazione che permettesse di registrare e pianificare le attività che svolgono i dipendenti durante l'arco della giornata lavorativa. Nel corso della stesura della tesi parlerò, nella prima parte, del mondo mobile e delle tecnologie di sviluppo, e nella seconda parte descriverò nello specifico la progettazione e l'implementazione dell'applicazione che ho realizzato, motivando le scelte che mi hanno portato ad adoperare le tecnologie utilizzate.

2. Le applicazioni cross-platform

2.1. *Comparazione nello sviluppo di app native e app multiplatforma*

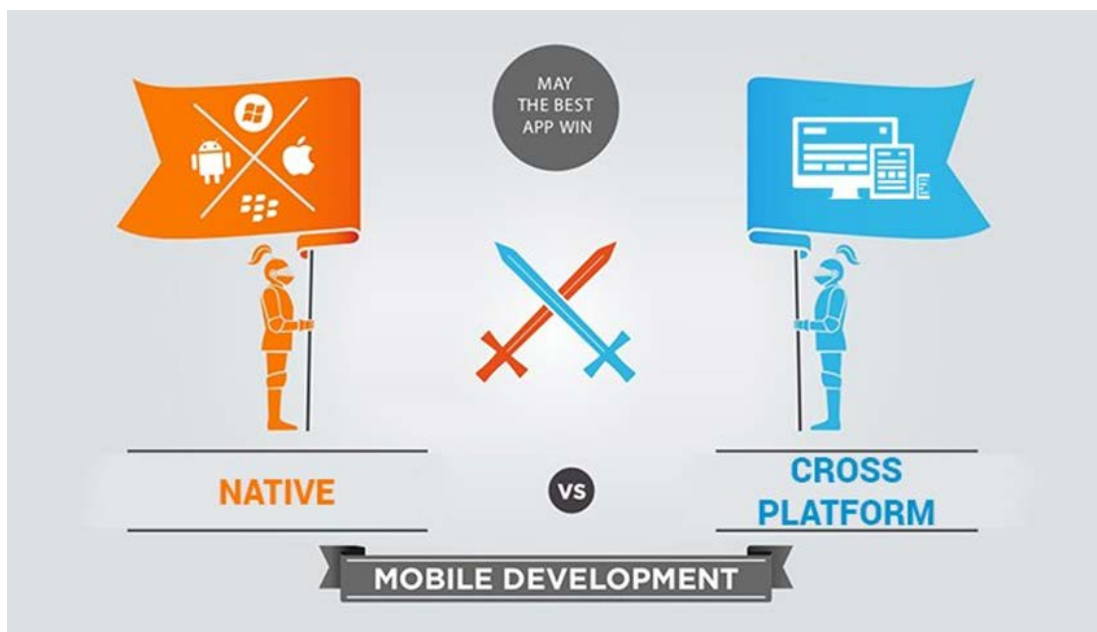
Sviluppo Nativo significa programmare e sviluppare un'app per una singola piattaforma, una programmazione che segue le linee guida della casa madre che ha creato l'ambiente di sviluppo ed il linguaggio di programmazione. Queste hanno dalla loro parte il vantaggio di poter sfruttare appieno le caratteristiche della piattaforma mobile e le funzionalità del dispositivo, di essere generalmente più efficienti e di avere un aspetto grafico integrato con quello dell'interfaccia grafica del sistema operativo. Le app native sono quindi consigliabili quando si ricerca una performance alta per un'applicazione con tanti elementi che sfruttano al massimo le potenzialità hardware del dispositivo (giochi, app con sistema di booking). Per intenderci, se vogliamo programmare e sviluppare app Android dobbiamo avere almeno le conoscenze basi di programmazione in JAVA, oppure in un altro linguaggio ad oggetti. Per iOS e Swift (linguaggi di programmazione per app iPhone/iPad) è necessario conoscere almeno un linguaggio di programmazione ad oggetti (meglio il C/C++/C#).

L'alternativa allo sviluppo nativo è quella di iniziare a programmare con linguaggi più semplici per il web come Javascript, Html e CSS per poi imparare ad utilizzare alcuni strumenti software per la creazione di app multiplatforma. Software, definiti in gergo tecnico "framework", che permettono di programmare una sola volta le app con linguaggi più semplici e ottenere in modo automatico la stessa app per i diversi dispositivi (Android, iOS, Window, ecc.). Infatti, ci penserà il compilatore del framework a creare le "APK" per ciascun dispositivo. La programmazione cross-platform è una modalità di sviluppo delle soluzioni software indipendente dal sistema operativo e dal produttore hardware, o per meglio dire che "funziona" su più di un sistema.

Le cross-platform hanno dei tempi di realizzazione dimezzati rispetto ad un app nativa, esse sono consigliabili per applicazioni che non richiedono determinate potenzialità hardware e che hanno dei tempi di consegna abbastanza limitati; queste

particolarità non escludono il fatto che un'applicazione cross-platform sia meno performante se si ricerca una determinata particolarità nell'applicazione.

In conclusione, non esiste una soluzione univoca per qualsiasi esigenza. In generale l'utilizzo degli strumenti di sviluppo nativi permette da sempre di ottenere i risultati migliori, una maggiore libertà espressiva e garanzia del risultato in termini di performance e user experience. In futuro molto probabilmente il linguaggio HTML5 (e le soluzioni ibride) guadagneranno maggiore importanza, ma oggi risultano utili per lo più nella creazione delle applicazioni semplici, che non abbiano requisiti importanti in termini di performance e di user experience (ad esempio, applicazioni B2B). Inoltre, l'adozione delle soluzioni cross-platform a volte ha svantaggi superiori al vantaggio economico che deriva dal loro utilizzo, che per questo motivo va attentamente ponderato.



2.2. Sviluppo nativo

Vediamo i principali vantaggi della programmazione nativa:

- Fornisce la migliore user experience all'utente, con la massima fluidità, responsività e velocità di interazione.
- Utilizza gli elementi nativi della piattaforma in termini di interfaccia utente, fornendo all'utente un ambiente che gli è già familiare.
- Ottiene la massima performance in termini di potenza di calcolo e di sfruttamento del processore.
- Sfrutta tutte le caratteristiche specifiche della piattaforma, sia in termini di software (librerie del produttore o di terze parti) che di hardware (del dispositivo o esterno)

La programmazione nativa comporta però anche degli svantaggi; eccone alcuni:

- È necessario riscrivere l'applicazione per ogni piattaforma che si intende supportare.
- Richiede conoscenza specifica dei vari linguaggi e piattaforme, e generalmente sviluppatori con skill specifici e più esperti.
- A causa dei punti precedenti può comportare un maggiore costo complessivo e l'allungamento dei tempi di sviluppo, in particolare dovendo gestire più piattaforme.
- Per la pubblicazione è richiesto generalmente di passare attraverso gli app store e sottostare ai relativi vincoli.

2.3. Sviluppo multi-platform

Ecco i principali vantaggi della programmazione multi-platform:

- Permette di sviluppare una sola volta l'applicazione, e di eseguirla su diverse piattaforme senza doverla riscrivere completamente.
- Permette di riutilizzare gli skill degli stessi sviluppatori che lavorano sulle tecnologie Web.
- È possibile creare applicazioni Ibride, installabili sul dispositivo smartphone o tablet come le app.
- Grazie ai punti precedenti, può permettere un risparmio di tempi e costi, specialmente per la creazione di applicazioni multiplatforma molto orientate ai contenuti.
- In caso di App completamente HTML5, non è necessario passare dagli app store, quindi non esistono vincoli di pubblicazione e gli aggiornamenti sono immediati.

Ovviamente la programmazione multi-platform comporta anch'essa degli svantaggi:

- Offre una user experience inferiore in termini di fluidità, responsività e velocità di interazione, specialmente sui dispositivi meno recenti o meno performanti.
- Non è adatto a progetti molto complessi o che richiedono elevate capacità di calcolo.
- Non fornisce strumenti specifici per gestire la frammentazione dei dispositivi, ed è più complesso nelle operazioni di debug.
- Anche se l'applicazione prodotta funziona su più piattaforme, possono essere necessari degli adattamenti, per esigenze di compatibilità, di aderenza alle specifiche software o di user interface
- Non permette di accedere a tutte le funzionalità del dispositivo, ma solo a un sottoinsieme di queste, a meno di non utilizzare wrapper e utilizzare estensioni native.

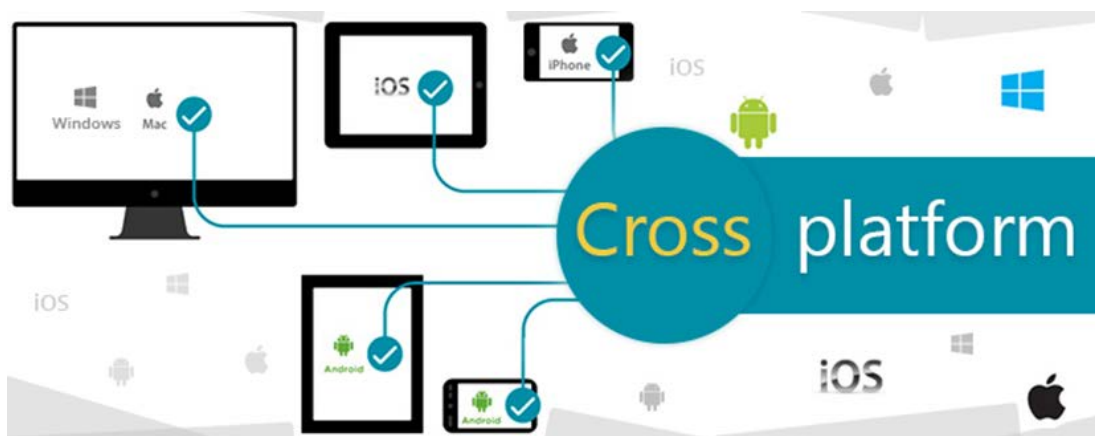
2.4. *Principali framework per lo sviluppo di app cross-platform*

Sviluppare un'app che possa essere eseguita sulle più comuni piattaforme mobile significa sviluppare applicazioni diverse con linguaggi differenti.

Le applicazioni ibride sfruttano il meglio dell'approccio nativo e di quello Web, consentendo di utilizzare lo stesso codice basato sulle tecnologie Web per le diverse piattaforme mobile. Tuttavia anche le applicazioni ibride possono soffrire dello stesso problema delle applicazioni Web: la mancata integrazione dell'aspetto grafico con la specifica piattaforma mobile.

Le applicazioni Web, sfruttando tecnologie standard, consentono di condividere lo stesso codice tra piattaforme mobile diverse. Dal momento però che si tratta sostanzialmente di siti Web ottimizzati per il mobile, le loro funzionalità non sono disponibili quando un dispositivo è offline. Inoltre, le applicazioni Web non possono accedere alle funzionalità del dispositivo, come ad esempio il file system, ed il loro aspetto grafico è generalmente non integrato con quello della piattaforma.

Tutti danno come risultato finale il classico slogan: “Scrivi una volta, esegui dappertutto”. Se voglio un'app non prettamente web (ma nessuno mi impedisce di interagire con il web, anzi!), o comunque voglio mantenere tutte le funzioni specifiche di ogni piattaforma, minimizzando il codice e magari scrivendo con un linguaggio solo, anche qui ho almeno 2 o 3 alternative papabili.



I software che permettono lo sviluppo APP cross-platform sono molti:
PhoneGap, Corona SDK, Xamarin, Appcelerator Titanium, ecc.

Adobe PhoneGap

Adobe PhoneGap è una soluzione cross-platform per sviluppare app iPhone, Android, Windows Phone ed altre importanti piattaforme mobile. Per utilizzarlo, basta caricare online i propri progetti scritti in HTML5, CSS e JavaScript e lasciare che il servizio compili l'applicazione. Sono disponibili vari piani di abbonamento, quello base è gratuito ma naturalmente per pubblicare applicazioni su App Store ed altri store di applicazioni è necessario pagare i relativi servizi.

Corona SDK

Corona SDK è un software che permette di sviluppare applicazioni per iOS e Android abbastanza semplice da usare. È basato su un linguaggio di programmazione denominato Luna che è più semplice da apprendere rispetto all'Object-C e il C+++ ed include tutto il necessario per creare, testare e pubblicare le app. Il software è a pagamento ma è disponibile in una versione di prova gratuita che permette di testarne tutte le funzioni per un periodo di 30 giorni. È disponibile per Windows e Mac OS X ma per creare app compatibili con iPhone, iPad ed iPod Touch occorre utilizzare necessariamente la versione per Mac OS X.

Xamarin

Xamarin è una piattaforma basata sul linguaggio C#, pensata per consentire principalmente lo sviluppo di applicazioni per iOS, Android e Windows, che possono essere compilate in modo distinto durante il deploy. Si tratta di un framework abbastanza avanzato, che consente agli sviluppatori l'utilizzo trasparente delle API native nell'app prodotta. Tramite Xamarin.Forms, gli sviluppatori hanno inoltre a disposizione un vasto numero di componenti per la realizzazione dell'interfaccia, in grado di adattarsi facilmente ai sistemi operativi mobile supportati. Xamarin offre la possibilità di eseguire il codice su un framework .NET, potendo simulare l'esecuzione dell'app sia su iOS che su Android. Risulta versatile soprattutto per

grandi team, in cui molte persone partecipano allo stesso progetto. L'intera piattaforma può essere utilizzata tramite l'IDE Microsoft Visual Studio, le cui funzionalità sono ben note anche a chi non si occupa di sviluppo mobile.

Appcelerator Titanium

Si tratta di un framework che consente di utilizzare linguaggi tipici dello sviluppo web, come JavaScript, XML e CSS, per generare app native per iOS, Android, Windows Phone e Tizen (è anche in grado di creare una versione web dell'app, che utilizza HTML5 e le suddette tecnologie per l'interfaccia e la business logic). Dal sito ufficiale è possibile ottenere l'IDE e tutti gli strumenti per iniziare lo sviluppo. I principali vantaggi di Titanium risiedono nella sua facilità d'uso, nonché nella possibilità di utilizzare moduli già pronti (che possono essere multiplatforma o specifici per un singolo sistema). Ciò è possibile grazie alla presenza di un Marketplace che offre una vasta gamma di moduli già pronti per soddisfare le esigenze più svariate. È possibile, inoltre, creare i propri moduli e condividerli. Inoltre, la possibilità di utilizzare i servizi cloud ACS (Appcelerator Cloud Services), e di integrarli facilmente all'interno delle app generate con Titanium, rende queste ultime facilmente orientabili al cloud, con la possibilità di gestire dati ed utenti tramite un backend già pronto e funzionale.

2.5. Il pattern Model-View-Controller

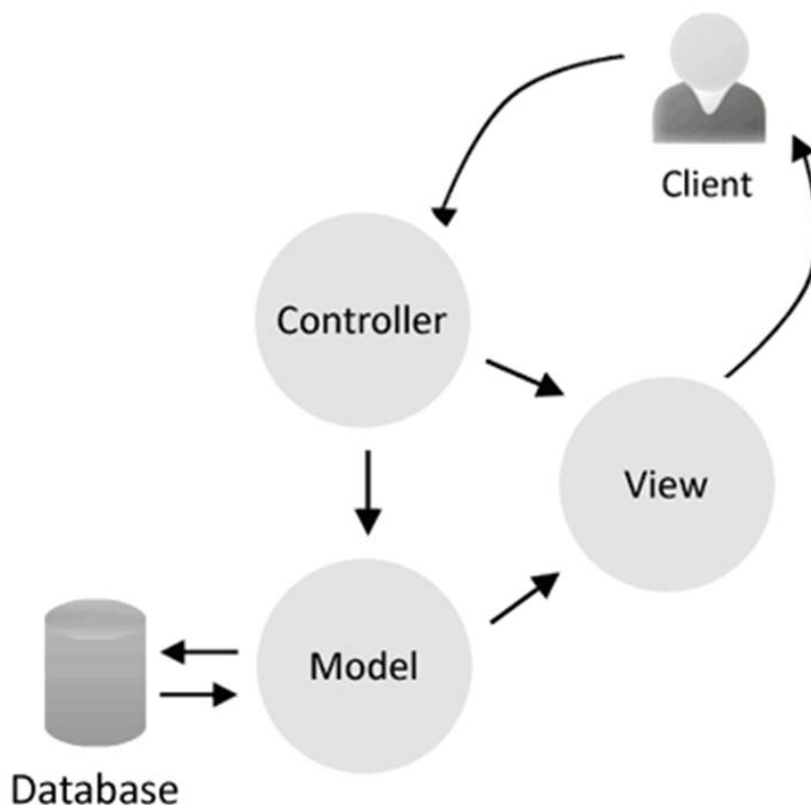
S Il Model-View-Controller (MVC, talvolta tradotto in italiano Modello-Vista-Controllo), in informatica, è un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti, in grado di separare la logica di presentazione dei dati dalla logica di business.

Questo pattern si posiziona nel livello di presentazione in una Architettura multi-tier. Il pattern è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- Il *model* fornisce i metodi per accedere ai dati utili all'applicazione;
- Il *view* visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;
- Il *controller* riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.

Questo schema, fra l'altro, implica anche la tradizionale separazione fra la logica applicativa (in questo contesto spesso chiamata "logica di business"), a carico del controller e del model, e l'interfaccia utente a carico del view.

I dettagli delle interazioni fra questi tre oggetti software dipendono molto dalle tecnologie usate (linguaggio di programmazione, eventuali librerie, middleware e via dicendo) e dal tipo di applicazione (per esempio se si tratta di un'applicazione web, o di un'applicazione desktop). Quasi sempre la relazione fra view e model è descrivibile anche come istanza del pattern Observer. A volte, quando è necessario cambiare il comportamento standard dell'applicazione a seconda delle circostanze, il controller implementa anche il pattern Strategy.



3. Tecnologie utilizzate nell'applicazione

In questa parte verranno mostrate e descritte tutte le tecnologie che ho utilizzato per la realizzazione dell'applicazione, illustrando all'inizio il significato e dopo il funzionamento ed il ruolo che esse hanno avuto nella programmazione dell'applicazione.

3.1 Ionic

Ionic è un framework che consente di sfruttare al meglio le tecnologie Web per creare applicazioni mobile con *look and feel* simile a quelle native e in questo ambito si rivela tra le soluzioni di maggior successo. Ionic è un insieme di API e tool pensati per lo sviluppo di app ibride: in altre parole app scritte in HTML+CSS+JavaScript che però hanno accesso a features tipiche dei dispositivi come fotocamera, accesso ai file, etc etc.

La parte HTML contribuisce allo scheletro dell'applicazione; grazie a questo strumento è possibile identificare le varie sezioni dell'applicazione. La parte CSS dà vita allo scheletro stesso; essa permette di personalizzare l'aspetto di ogni pagina usata nell'applicazione. La parte JavaScript è il vero e proprio collante tra applicazione e grafica. Grazie ad essa è possibile usufruire delle diverse features realizzate in quest'app. Per il codice JavaScript si è utilizzato il framework AngularJS. Angular JS non è altro che il "supereroe" di JavaScript. Questo potente strumento, ideato da Google, prevede un meccanismo che consente di definire le view in file separati (partial templates) e gestirle all'interno di uno schema comune, detto layout template.

Per un utilizzo performante di questi strumenti sono stati affiancati dei framework che ne innalzano la potenza e la performance.

Il framework è realizzato dal team Drifty e gli stessi autori affermano che Ionic può essere considerato “*il Bootstrap per le applicazioni ibride*”: come Bootstrap ha significato un punto di riferimento per la creazione di siti e applicazioni Web responsivi, Ionic vuole essere il punto di riferimento per la creazione di applicazioni mobile ibride con un aspetto il più possibile vicino ai componenti della piattaforma nativa.

I loro autori hanno raccolto le migliori *best practice* per lo sviluppo di interfacce mobile con tecnologie Web e le hanno codificate in questo framework, evitando quindi che ogni sviluppatore riparta da zero nello sviluppo dell’interfaccia di una nuova applicazione.

L’interfaccia visiva che si sviluppa con Ionic è molto facile da utilizzare, si integra perfettamente con il look-and-feel delle varie piattaforme ed è molto facile da personalizzare secondo il design della nostra app.

Come avremo modo di vedere, lo sviluppo di applicazioni ibride con Ionic è semplificato dalla disponibilità di un vero e proprio ambiente di sviluppo che consente di creare lo scheletro iniziale di un’applicazione, di compilarla per una piattaforma mobile e di effettuare il debug all’interno di un browser. A ciò si aggiunge la licenza MIT che consente a chiunque di utilizzare liberamente il framework e i suoi sorgenti.

Ionic si basa su dei pilastri come Apache Cordova (per l’interfacciamento tra le chiamate native del sistema ospitante e javascript) e AngularJS (per tutto ciò che riguarda la logica dell’applicazione).

Da un punto di vista tecnico, il framework utilizza AngularJS come motore per definire i componenti dell’interfaccia grafica. In sostanza, Ionic è costituito da un insieme di direttive Angular riutilizzabili ed eventualmente espandibili per disegnare l’interfaccia grafica con un approccio dichiarativo.

L'architettura di Apache Cordova si presenta come una sorta di contenitore di applicazione Web eseguita localmente. L'interfaccia grafica di un'applicazione Cordova é infatti costituita da una Web view che occupa l'intero schermo del dispositivo e all'interno della quale viene visualizzato l'HTML ed il CSS ed eseguito il codice JavaScript. Tramite JavaScript è possibile accedere ad un ricco insieme di API che interfacciano l'applicazione Web con le funzionalità della piattaforma ospite.



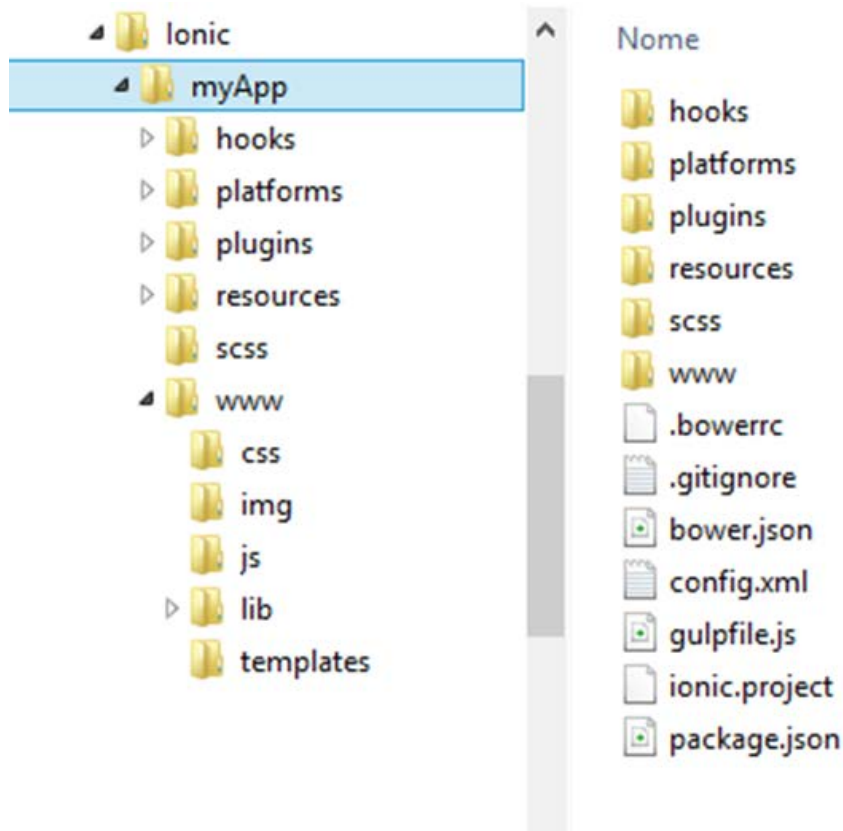
3.2 Struttura di un progetto Ionic

La creazione di un nuovo progetto Ionic è strettamente correlata alla creazione di un progetto Cordova. Infatti, come avremo modo di vedere, Ionic fa da wrapper a diversi comandi Cordova per la gestione dell'ambiente di sviluppo.

Digitando il seguente comando in una cartella di lavoro creeremo l'infrastruttura necessaria per sviluppare la nostra app ibrida myApp con un layout grafico supportato da Ionic:

```
ionic start myApp
```


Ecco il contenuto della cartella myApp creata in seguito all'esecuzione del comando:



Oltre ad alcuni file e cartelle di configurazione di Cordova, troviamo alcuni elementi specifici di Ionic. In particolare abbiamo:

- il file ionic.project che contiene la configurazione di Ionic per il progetto corrente
- la cartella scss che contiene il codice SASS per la generazione dei CSS
- la cartella www che conterrà il codice HTML, JavaScript, CSS e tutto l'occorrente per definire la nostra app ibrida.

La cartella www

Il contenuto della cartella *www* dipende dal template di progetto che utilizziamo per sviluppare la nostra applicazione, come vedremo più avanti. La sua struttura standard prevede la presenza delle seguenti cartelle:

Cartella	Cosa contiene
<i>css</i>	I file CSS derivanti dalla compilazione del codice SASS di Ionic ed eventuali altri CSS specifici per la nostra applicazione.
<i>img</i>	Le risorse grafiche.
<i>lib</i>	Il codice di <i>Ionic</i> e quello di eventuali librerie di terze parti installate tramite <i>bower</i> .
<i>js</i>	Il codice JavaScript della nostra applicazione. In essa troviamo una serie di file che propongono uno schema per la definizione dell'applicazione AngularJS, ma l'utente può cambiare la struttura proposta in base alle proprie esigenze.
<i>template</i>	Pensata per ospitare i template delle view Angular.

In particolare, nella cartella js sono presenti i file:

- app.js che contiene la configurazione e l'inizializzazione dell'applicazione Angular
- controllers.js destinato a contenere i controller utilizzati nell'app
- services.js e directives.js sono pensati per ospitare rispettivamente i servizi e le direttive dell'applicazione.

Infine, il file *index.html* contiene il markup della pagina all'interno della quale saranno visualizzate le view dell'applicazione.

3.3 *Ionic View*

Ionic View è un servizio che consente di avere un approccio unificato per la preview ed il beta-testing di un'applicazione mobile sviluppata con Ionic Framework.

L'approccio è immediato: una volta realizzata la nostra applicazione possiamo eseguire all'interno della sua cartella il seguente comando tramite Command Line Interface:

```
ionic upload
```

L'applicazione sarà caricata sui server di Ionic ed a conclusione del processo di upload verrà visualizzato un messaggio con un codice alfanumerico che rappresenta l'identificatore dell'app su Ionic View.

A questo punto è sufficiente comunicare questo identificatore alle persone coinvolte nella valutazione o nel test dell'applicazione mobile.

Queste persone dovranno scaricare l'app *Ionic View* dagli store di Apple o di Google in base al sistema operativo del proprio dispositivo, registrarsi o autenticarsi ed inserire il codice identificativo dell'app. L'applicazione verrà scaricata e potrà essere utilizzata come se fosse stata scaricata dagli store ufficiali.

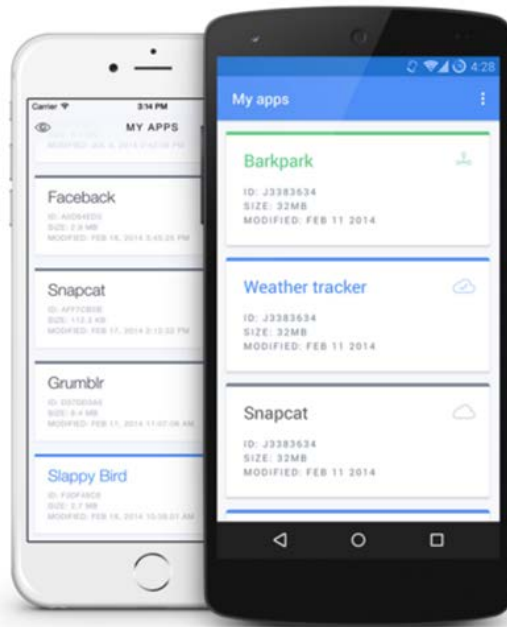
Senza nulla togliere agli ambienti di test di Apple e Google, l'approccio di *Ionic View* è *molto semplice* in quanto con un solo comando consente di mettere a

disposizione la propria applicazione su piattaforme diverse velocizzando l'attività di valutazione e test esterni.



Share your apps with the world

Ionic View makes it easy to share your [Ionic](#) and Cordova apps with clients and testers around the world, all without ever going through the App Store.



3.4 Apache Cordova

Nello sviluppo di applicazioni mobile ibride Cordova è senza dubbio uno dei framework più noti insieme al suo derivato PhoneGap. Si tratta sostanzialmente di framework di sviluppo che consentono di generare applicazioni native per le diverse piattaforme mobile che incorporano applicazioni Web.

Grazie a questi strumenti, è possibile creare la logica e interfaccia utente dell'applicazione sfruttando HTML, CSS e JavaScript, accedere alla funzionalità della piattaforma mobile tramite apposite API e creare pacchetti per ciascuna piattaforma.

Grazie al Framework Apache Cordova è possibile tramite questi strumenti pubblicare per più Stores con l'utilizzo di un solo codice sorgente.

Grazie ad Apache-Cordova la parte di distribuzione delle applicazioni per i diversi sistemi non cambia nulla rispetto alla distribuzione con app Nativa. Si prenderà in considerazione ogni applicazione con il rispetto store di appartenenza e si procederà come con le app-native, Android usa Play Store, dopo aver creato un account da developer ed aver pagato 25 dollari per l'acquisizione, per quanto riguarda iOS il prezzo da pagare per l'account da sviluppatore è di 100 dollari annuali. Una volta acquistati gli accounts per i diversi Stores, bisogna caricare i file sorgenti e aspettare che i sistemi di Stores accettano il file. Passeranno alcune ore prima della pubblicazione sui rispettivi Stores. Tutte le applicazioni infatti potranno essere rilasciate nel proprio Store di appartenenza senza particolari cambiamenti di configurazione

La pubblicazione nei diversi Stores naturalmente cambia da dispositivo a dispositivo. Nel caso specifico si è preferito utilizzare il 98% dei dispositivi in circolazione ovvero quelli compresi di sistema operativo Android o iOS.

3.5 AngularJS

AngularJS è un framework, una infrastruttura per la creazione di applicazioni composta da un insieme di funzionalità. Fornisce tutto quanto occorre per creare applicazioni moderne che sfruttano le più recenti tecnologie, come ad esempio le Single Page Application, cioè applicazioni le cui risorse vengono caricate dinamicamente su richiesta, senza necessità di ricaricare l'intera pagina.

Per raggiungere questo obiettivo, AngularJS da un lato esalta e potenzia l'approccio dichiarativo dell'HTML nella definizione dell'interfaccia grafica, dall'altro fornisce strumenti per la costruzione di un'architettura modulare e testabile della logica applicativa di un'applicazione.

Questi concetti, probabilmente ora un po' astratti, dovranno essere tenuti presenti per comprendere a pieno le potenzialità di AngularJS evitando di utilizzarlo (male) come un'ennesima libreria JavaScript.

Tra le principali funzionalità a supporto dello sviluppo di tali applicazioni segnaliamo:

- il binding bidirezionale (two-way binding)
- la dependency injection
- il supporto al pattern MVC
- il supporto ai moduli
- la separazione delle competenze
- la testabilità del codice
- la riusabilità dei componenti

AngularJS offre allo sviluppatore un approccio orientato più al "cosa fare" che al "come farlo" lasciando quindi al *parser* il compito di raggiungere gli obiettivi descritti. Il risultato di un simple approccio, come è facile intuire, è quello di avere applicazioni molto snelle e composte da poche righe di codice. Questo framework opera manipolando ed estendendo il codice HTML della pagina al fine di presentare contenuti dinamici attraverso un data binding a due vie che consente una sincronizzazione automatica di *models* e *views*: ogni modifica al modello viene riportata automaticamente nella vista (mediante un aggiornamento delle *espressioni*) ed allo stesso tempo ogni modifica nella vista si riflette sul modello, il tutto senza che lo sviluppatore debba preoccuparsi di manipolare direttamente il DOM.

3.6 *PhpMyAdmin*

PhpMyAdmin è un'applicazione web scritta in PHP, distribuita con licenza GPL, che consente di amministrare un database MySQL tramite un qualsiasi browser. L'applicazione è indirizzata sia agli amministratori del database, sia agli utenti. PhpMyAdmin permette di creare un database da zero, creare le tabelle ed eseguire operazioni di ottimizzazione sulle stesse. Presenta un feedback sulla creazione delle tabelle per evitare eventuali errori. Sono previste delle funzionalità per l'inserimento dei dati (popolazione del database), per le query, per il backup dei dati, ecc.

L'amministratore ha anche a disposizione un'interfaccia grafica per la gestione degli utenti: l'interfaccia permette l'inserimento di un nuovo utente, la modifica della relativa password e la gestione dei permessi che l'utente ha sul database.

PHP (Hypertext Preprocessor acronimo di "Personal Home Page") è un linguaggio di scripting interpretato, originariamente concepito per la programmazione di pagine web dinamiche. Questo strumento è stato utilizzato come portale tra il client (in questo caso l'app) ed il server; è stato utilizzato anche per l'inserimento dei form nel database MySQL. Esso è di pratico utilizzo con funzioni di base specifiche per la connessione tra client e server.

3.7 *GitHub*

GitHib è un servizio di hosting web per progetti di sviluppo software che vogliono usare il sistema di controllo di versione Git (ideato da Linus Torvalds). GitHub offre sia piani commerciali che account gratuiti per progetti open source.

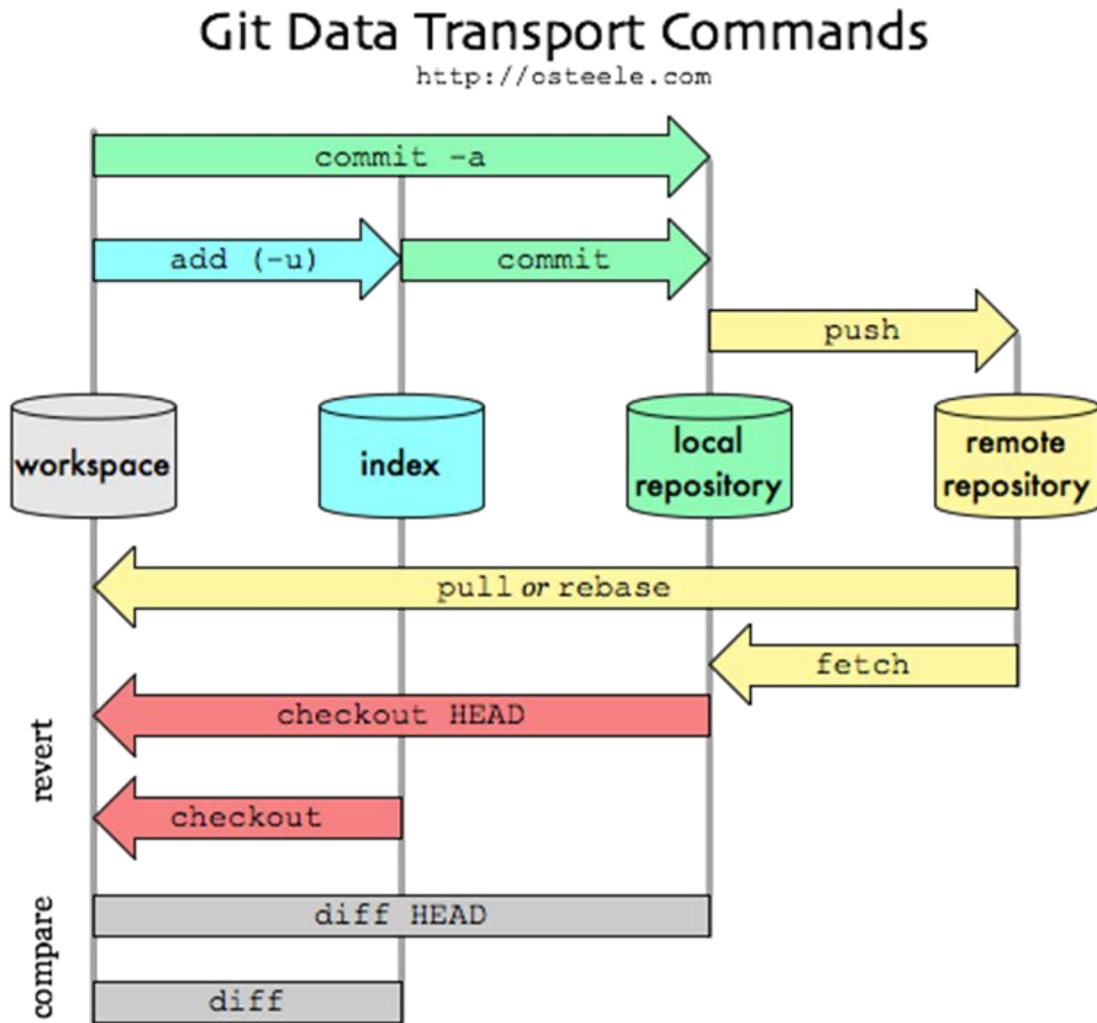
I progetti su Github nascono per consentire la condivisione e la collaborazione, offrendo quindi ai programmatori che lo desiderano di contribuire alla realizzazione di incredibili progetti. Come prima cosa è necessario creare una copia del progetto (e quindi del repository) sul proprio account Github.

Per farlo è sufficiente accedere alla pagina Github del progetto al quale si desidera contribuire

Un sistema di controllo versione distribuito o decentralizzato (o DVCS da Distributed Version Control System) permette di tenere traccia delle modifiche e delle versioni apportate al codice sorgente di un software, senza la necessità di dover utilizzare un server centrale.

Con questo sistema gli sviluppatori possono collaborare individualmente e parallelamente da offline su di un proprio ramo (branch) di sviluppo, registrare le proprie modifiche (commit) ed in seguito condividerle con altri o unite (merge) a quelle di altri, il tutto senza bisogno del supporto di un server.

Questo sistema permette diverse modalità di collaborazione, proprio perché il server è soltanto un mero strumento d'appoggio.



4 Progettazione dell'applicazione

In questo capitolo parlerò della fase precedente a quella realizzativa, esponendo quali architetture sono state utilizzate e di come vengono implementate nell'applicazione. Questa fase è molto importante perché stabilisce tutto quello che andrà ad essere sviluppato successivamente.

4.2 Applicazione di rete

Il cuore dello sviluppo delle applicazioni di rete è costituito dalla compilazione dei programmi che sono eseguiti dai sistemi terminali e comunicano tra loro via rete. Prima di occuparci della codifica software, che è significativamente diversa dall'architettura di rete. Per lo sviluppatore, l'architettura di rete è fissata e fornisce alle applicazioni uno specifico insieme di servizi, ma il suo compito è progettare l'architettura dell'applicazione e stabilire la sua organizzazione.

4.3 Architettura Client-Server

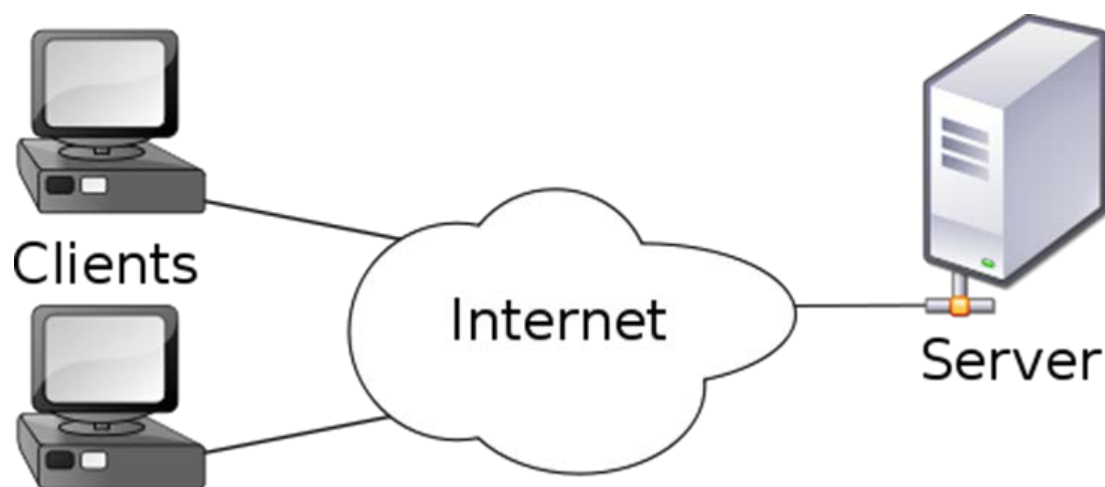
Nell'architettura client-server c'è un host sempre attivo, chiamato server, che risponde alle richieste di servizio di molti altri host, detti client. I client possono essere sempre o talvolta attivi. Il software client in genere è di limitata complessità, limitandosi normalmente ad operare come interfaccia verso il server. In generale nel campo informatico il termine client indica una componente che accede ai servizi o alle risorse di un'altra componente, detta server. In questo contesto si può quindi parlare di client riferendosi all'hardware o al software.

Un computer collegato ad un server tramite rete locale o geografica, ed al quale richiede uno o più servizi, utilizzando uno o più protocolli di rete è un esempio di client hardware.

Un programma di posta elettronica è un esempio di client software

Il software server, oltre alla gestione logica del sistema, deve implementare tutte le tecniche di gestione degli accessi, allocazione e rilascio delle risorse, condivisione e sicurezza dei dati o delle risorse.

Ad esempio un server di posta elettronica è paragonabile ad un qualunque ufficio postale. Gli utilizzatori per accedere via client alla loro cassetta di posta elettronica devono essere stati autorizzati. In modo analogo un utente deve possedere la chiave della cassetta sita presso un ufficio postale dalla quale vuole prelevare la corrispondenza.



4.4 Server

Il server è stato progettato per interagire con il database offrendo i seguenti servizi:

- Get sul database:

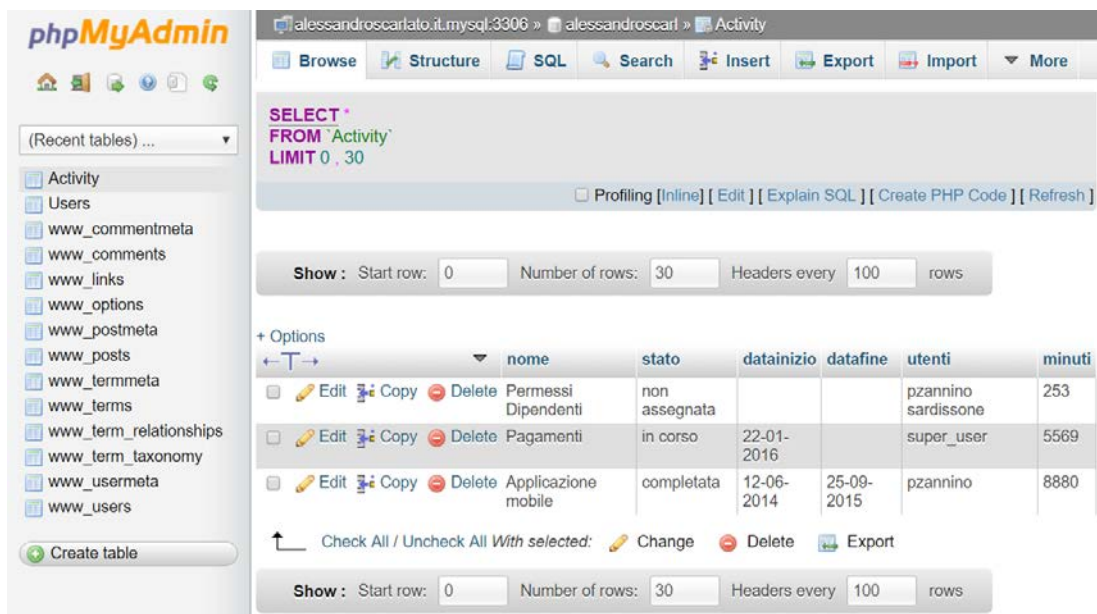
- *Controllo sull'accesso*
- *Tabella dei profili*
- *Tabella delle attività*

- Post sul database:

- *Modifica della password*
- *Aggiunta e modifica di un utente*
- *Aggiunta e assegnazione di un'attività*

4.5 Database

Ho scelto di appoggiarmi allo spazio web, one.com. Avendo già precedentemente comprato uno spazio web per il mio sito personale, volevo sfruttare i servizi che questo web hosting offriva. Uno di questi era appunto il database MariaDB(MySQL), che abbinato con il linguaggio di scripting PHP consente di comunicare e interrogare il database. PHP e MariaDB(MySQL) sono le principali tecnologie web open source, ovvero il loro codice sorgente è disponibile gratuitamente agli sviluppatori di tutto il mondo che possono utilizzarlo gratuitamente.



The screenshot shows the phpMyAdmin interface for a MySQL database. The left sidebar lists various tables, including 'Activity'. The main area displays the 'Activity' table with the following columns: 'nome', 'stato', 'datainizio', 'datafine', 'utenti', and 'minuti'. The table contains three rows of data:

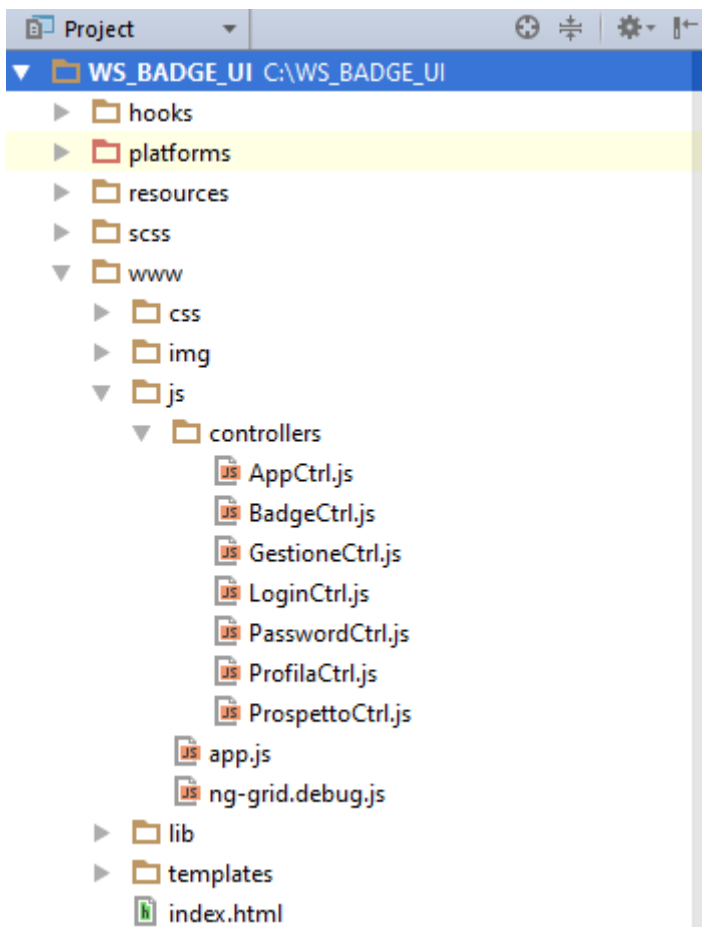
	nome	stato	datainizio	datafine	utenti	minuti
<input type="checkbox"/>	Permessi Dipendenti	non assegnata			pzannino sardissone	253
<input type="checkbox"/>	Pagamenti	in corso	22-01-2016		super_user	5569
<input type="checkbox"/>	Applicazione mobile	completata	12-06-2014	25-09-2015	pzannino	8880

5 Implementazione dell'applicazione

In questo capitolo verrà presentata l'applicazione che ho realizzato, sulla quale ho lavorato in questi mesi per l'azienda. L'applicazione è stata sviluppata con i sopracitati Framework: Ionic, AngularJs, Apache Cordova. PhpMyAdmin. Saranno espressi in maniera dettagliata tutte le informazioni e le funzionalità dell'applicazione.

5.2 Sviluppo dell'applicazione

L'applicazione è stata creata utilizzando come IDE di supporto WebStorm sviluppata dall'azienda JetBrains. Come mostra la figura, ci sono diverse cartelle che compongono il progetto.



La cartella *hooks* rappresenta il contenitore degli script speciali che potrebbero essere aggiunti all'applicazione per costruire il sistema per personalizzazione dei comandi di cordova. Può essere utile per i grandi progetti che richiedono processi di modifica del codice sorgente automatizzati ma normalmente è inutilizzata.

La cartella *platform* contiene tutte quei file che servono a far funzionare l'applicazione su una determinata piattaforma. In generale, non c'è bisogno di lavorare in questa directory a meno che non si sta facendo l'hacking nativo personalizzato o, eventualmente, di inviare la nostra applicazione in produzione.

Il comando per attivare la piattaforma Android è:

```
ionic platform add android
```

Il comando per attivare la piattaforma iOS è:

```
ionic platform add ios
```

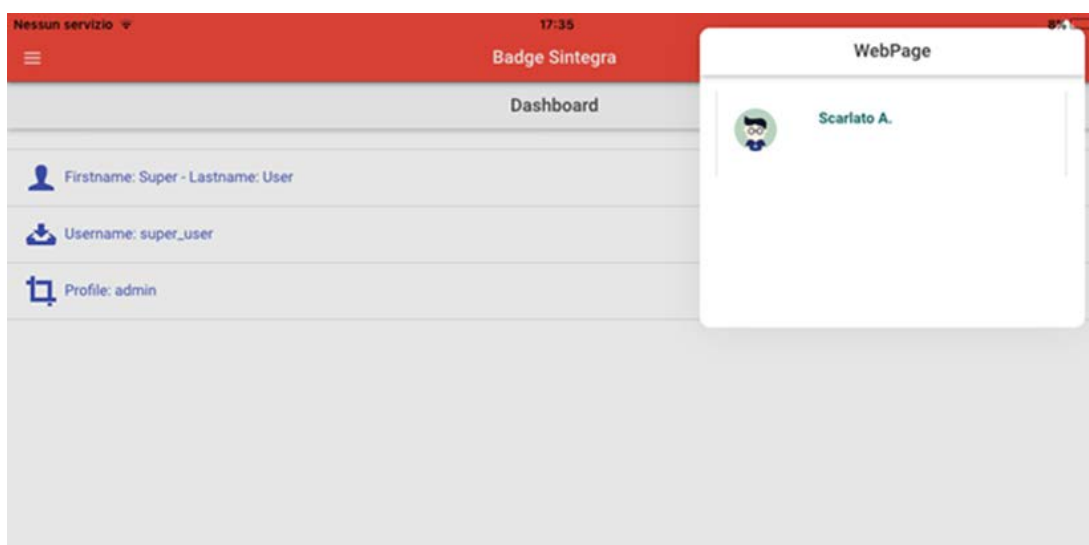
La cartella *scss* che contiene il codice SASS per la generazione dei CSS e dove si gestisce il look'n'feel dell'applicazione. L'aspetto grafico predefinito è ispirato al look and feel di iOS7, ma è possibile effettuare delle personalizzazioni effettuando l'override del CSS e/o impostando variabili e mixin di SASS.

I file prettamente di interesse per lo sviluppo dell'applicazione si trovano all'interno della cartella "*www*", qui si trovano tutti i file HTML, CSS e JavaScript. Il file *index.html* indica la pagina iniziale dell'applicazione, come nelle applicazioni web. Nell' head di questo file troviamo tutti i riferimenti alle librerie che verranno caricate, e che si trovano nella cartella *lib*. La cartella *css* contiene i tag per definire la formattazione dei documenti HTML. La cartella *img* serve da contenitore per tutte le immagini caricate nel progetto. Nella cartella *js* ci saranno tutti i file Javascript: il file *app.js*, consente di definire il passaggio tra una schermata e l'altra di un'applicazione come le transizioni di una macchina a stati; il contenuto della cartella *controllers* assolve il ruolo di creare un collegamento tra il model (la fonte dei dati) e la view (la presentazione), definendo un controller, in altre parole,

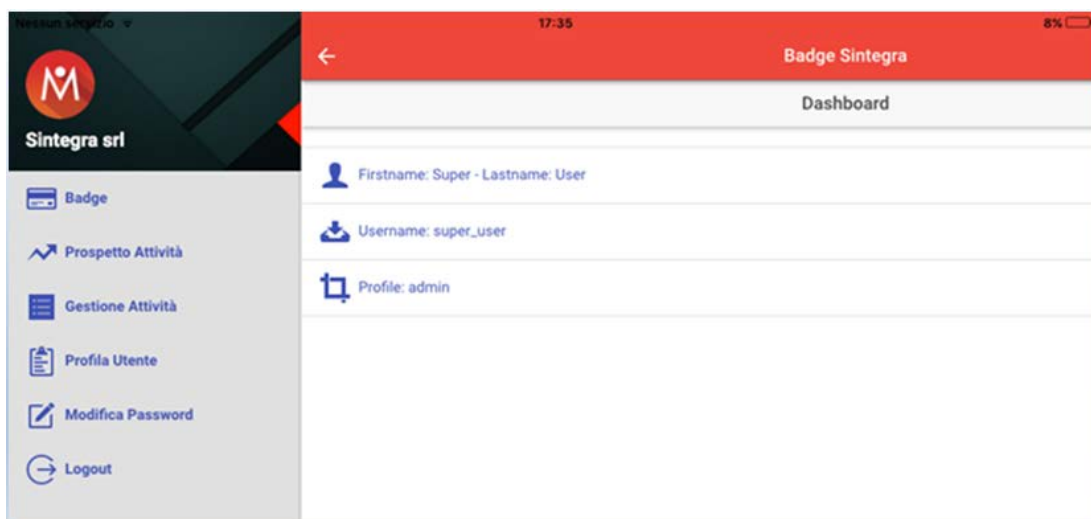
possiamo regolare il modo in cui i dati (grezzi) confluiscono all'interno del livello di presentazione. In AngularJS un controller viene definito come un semplice oggetto Javascript. La cartella lib è dove le librerie utilizzate possono essere collocate. Ionic e altre librerie che usi possono essere collocati. Infine troviamo la cartella templates pensata per ospitare i template delle view Angular.

5.3 Panoramica dell'applicazione

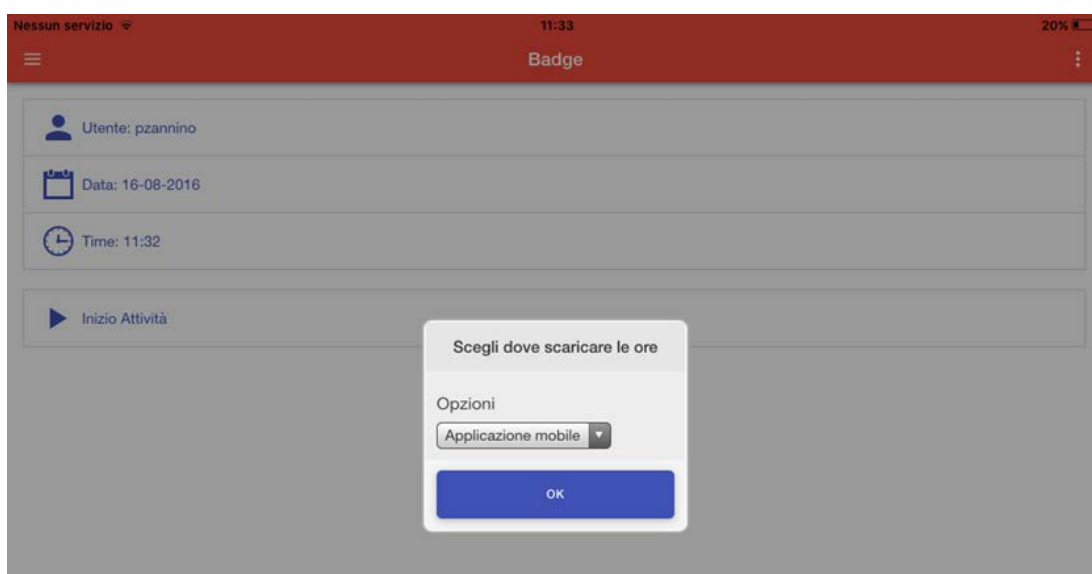
Quando l'applicazione viene lanciata, si presenta in questo modo: a sinistra abbiamo una left-bar, dove ci saranno tutti quei servizi che verranno mostrati e abilitati in base alla figura aziendale loggata; a destra abbiamo una right-bar con un link al mio sito personale, sviluppato in word-press e che quindi si adatta anche esso al format del dispositivo, infine nella parte centrale abbiamo la schermata della login, dove è previsto l'inserimento di username e password.



Una volta compilati i campi, l'utente vedrà una pagina di dashboard, dove verranno visualizzati le informazioni dell'utente: nome, cognome, username e il profilo. Abbiamo tre figure aziendali e ad ognuna di essa sono assegnati dei servizi: utente generico, manager, amministratore.



La prima figura aziendale è l'utente. L'utente una volta loggato, ha la possibilità di modificare la propria password, visionare le attività che sta svolgendo e infine usare il badge. Quando il dipendente inizia la sua giornata lavorativa, allora fa il "badge" per iniziare l'attività. Dopo di che può stoppare l'attività. Qui ci saranno una lista per motivare l'azione, il dipendente potrebbe voler andare in pausa pranzo, oppure richiedere un permesso, o proprio finire la giornata lavorativa. In questo caso l'utente dovrà scegliere su quale attività scaricare le ore.



Un' altro ruolo è quello del manager. Il manager oltre ad avere i servizi dell'utente, cioè modifica della password, prospetto delle attività, e del badge, può anche gestire le attività. Si presenterà una tabella dove ci saranno tutte le attività aziendali con i rispettivi stati. Un'attività può essere: non assegnata, in questo caso sarà di colore rosso; in corso, in questo caso sarà di colore giallo; o completata, in questo caso di colore blu. Nella riga di ogni attività, inoltre è possibile conoscere la data di inizio, la data fine (se è stata completata), gli utenti assegnati per quella specifica attività, e i minuti totali che gli sono stati dedicati. Oltre ad avere visione delle attività, il manager può interagire con la tabella delle attività. In particolare potrà: assegnare gli utenti ad una specifica attività, aggiungere un'attività, infine esportare la tabella in excel o in pdf.

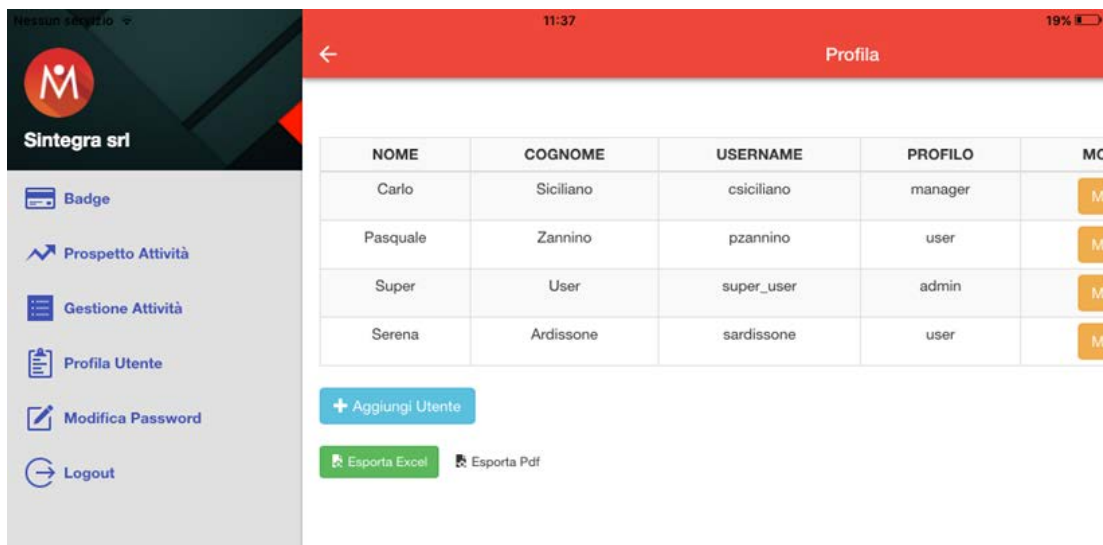


NOME ATTIVITÀ	STATO	DATA INIZIO	DATA FINE	UTENTI ASSEGNATI	MINUTI LAVORATI	ASSEGNA
Permessi Dipendenti	non assegnata			pzannino sardissone	253	Assegna
Pagamenti	in corso	22-01-2016		super_user	5568	Assegna
Applicazione mobile	completata	12-06-2014	25-09-2015	pzannino	88801	Assegna

+ Aggiungi Attività

Esporta Excel Esporta Pdf

L'ultimo modello aziendale è l'amministratore. Questo personaggio ha accesso a tutte le funzionalità dell'applicazione, quindi oltre ad avere l'accessibilità ai comandi del manager, cioè modifica della password, prospetto delle attività, della gestione delle attività e del badge, può anche gestire i profili dei dipendenti dell'azienda. Verrà mostrata una tabella con i rispettivi nomi, cognomi, username, e profilo dei dipendenti. L'amministratore potrà modificare alcuni campi del dipendente, quali il nome, il cognome, e il profilo; ha anche la possibilità di resettare la password dei dipendenti; potrà aggiungere un utente alla tabella; e, come nella gestione dell'attività, esportare la tabella dei profili in excel o in pdf.

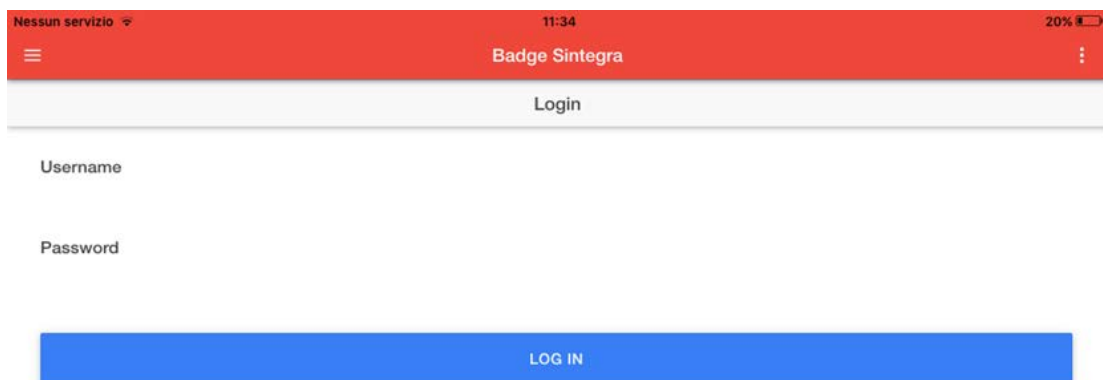


5.4 Principali funzionalità

Le principali funzionalità dell'applicazione sono:

Login

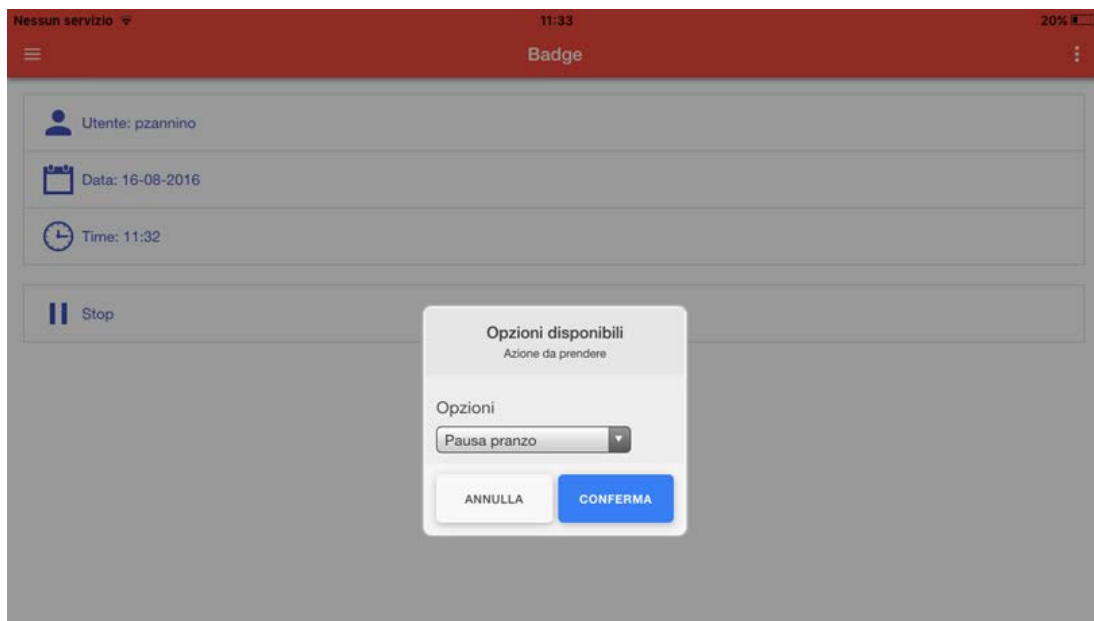
Form di inserimento di username e password per accedere alle funzionalità dell'applicazione.



Badge

Usato per iniziare la giornata lavorativa, oltre a visualizzare il nome dell'utente, la data e l'orario corrente, si vedrà visualizzato un pulsante start. Una volta premuto, si inizierà a conteggiare il tempo. In seguito, sarà un pulsante stop, una volta premuto le

possibili scelte saranno: pausa pranzo, l'utente utilizza l'unica pausa nell'arco della giornata per fare una pausa; richiesta permesso, l'utente richiedere un permesso scegliendo il motivo; scarico ore, questa funzionalità permette di scaricare le ore svolte, fino a questo momento, su una delle attività che sono assegnate all'utente.



Prospetto attività

Mostra le attività che si stanno svolgendo per l'azienda, e il conteggio dei minuti che l'utente gli ha dedicato.

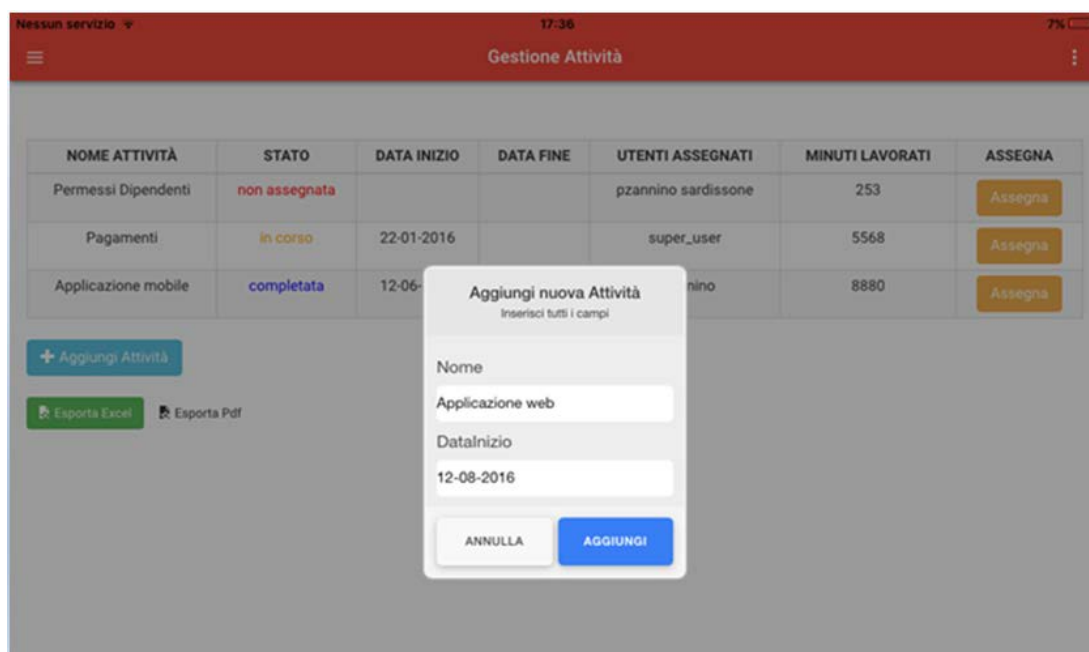
The screenshot shows the 'Prospetto Attività' app interface. The status bar displays 'Nessun servizio', '11:36', and '19%'. The app header is red with the title 'Prospetto Attività'. Below the header, there is a table with three columns: 'NOME ATTIVITÀ', 'STATO', and 'MINUTI LAVORATI'. The table contains one row of data.

NOME ATTIVITÀ	STATO	MINUTI LAVORATI
Pagamenti	in corso	5568

Gestione attività

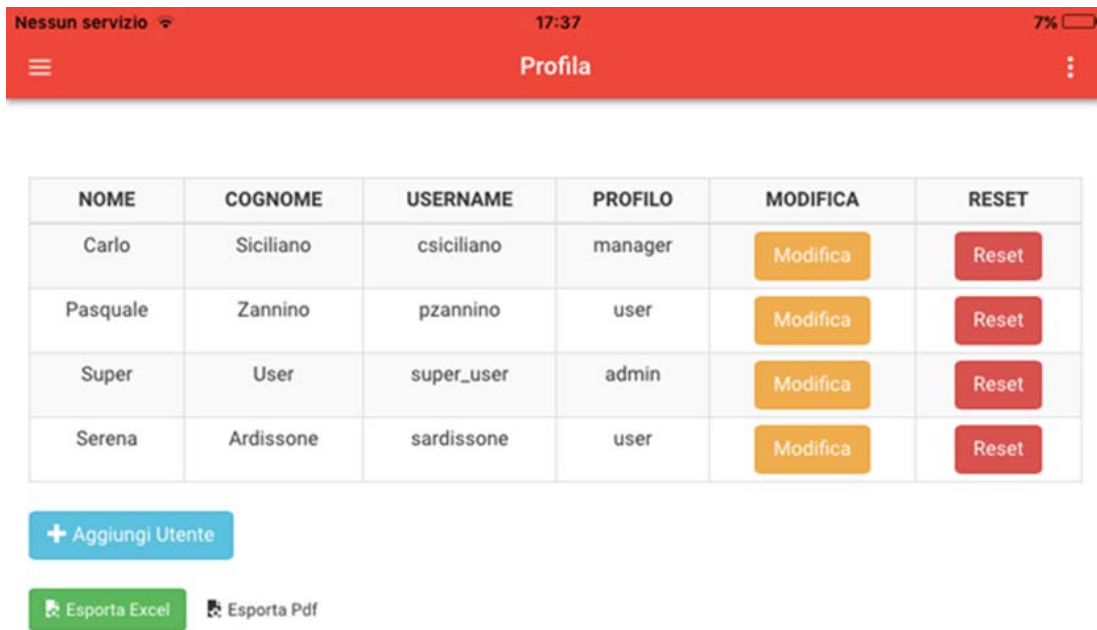
Mostra in una tabella le attività che si stanno svolgendo in azienda, e per ognuna di essa viene mostrato il nome, la data di inizio, l'eventuale data di fine, il conteggio delle ore, e i dipendenti che sono stati assegnati a quella determinata attività. Inoltre

è possibile modificare e interagire con la tabella. L'utente potrà: aggiungere una nuova attività, specificando il nome e la data di inizio; assegnare determinati utenti ad una specifica attività; infine esportare la tabella in formato excel o in formato pdf.



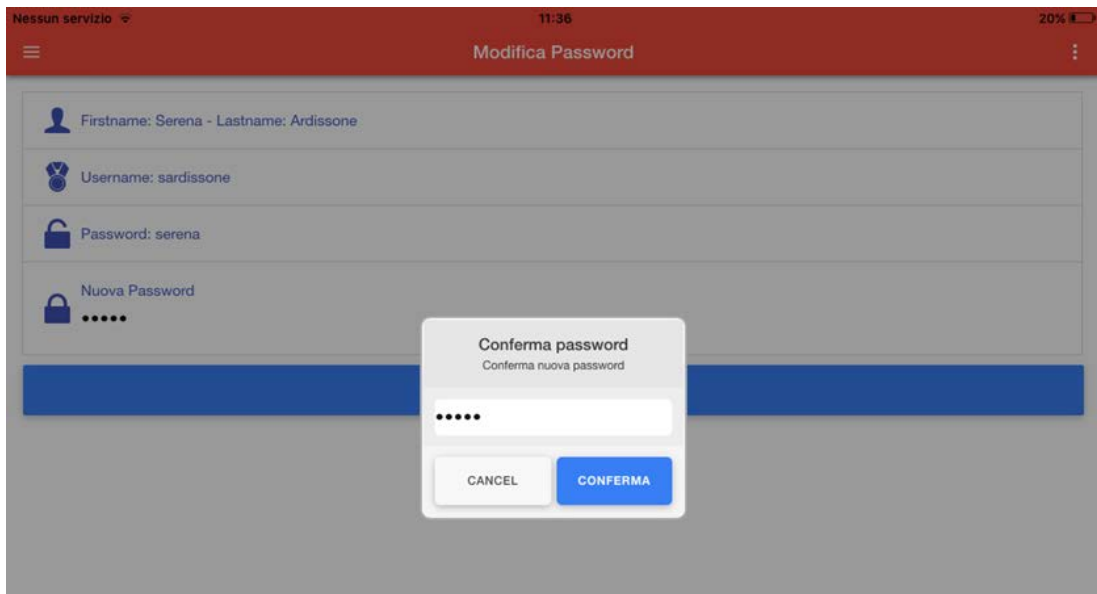
Profilo utente

In questa pagina vengono visualizzati tutti i dipendenti dell'azienda con il proprio nome, cognome, username, e il tipo di profilo. Anche qui possiamo interagire con la tabella: il tasto aggiungi utente, permette di aggiungere un utente al database, sarà necessario specificare tutti i campi richiesti; il bottone modifica, permette di modificare alcune informazioni relative all'utente selezionato, il tasto reset, resetta un valore di default la password dello user selezionato, anche qui come nella gestione delle attività abbiamo la possibilità di esportare la tabella in formato excel e pdf.



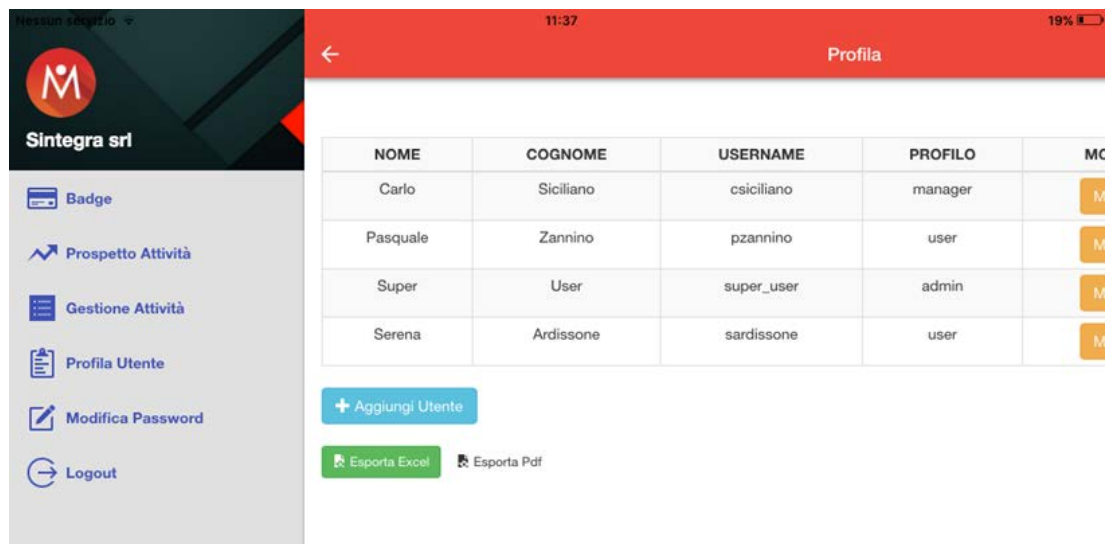
Modifica password

Pagina usata per la modifica della password. Dopo aver specificata la nuova password, verrà visualizzato un pop-up di conferma, le password dovranno essere identiche per apportare la modifica.



Logout

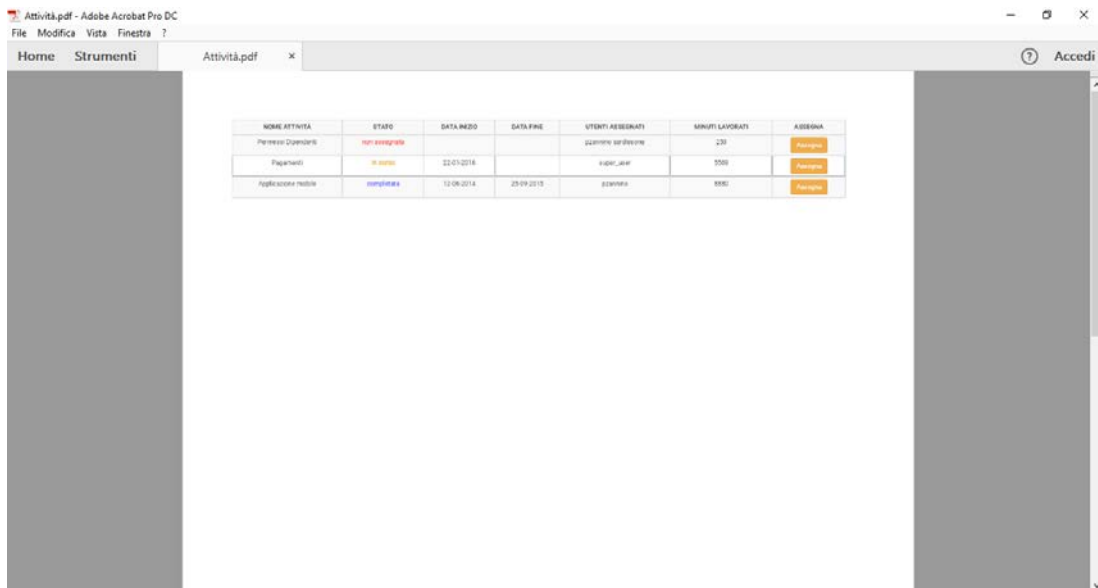
Usata per fare il logout delle credenziali. Si ritorna allo stato iniziale dell'applicazione.



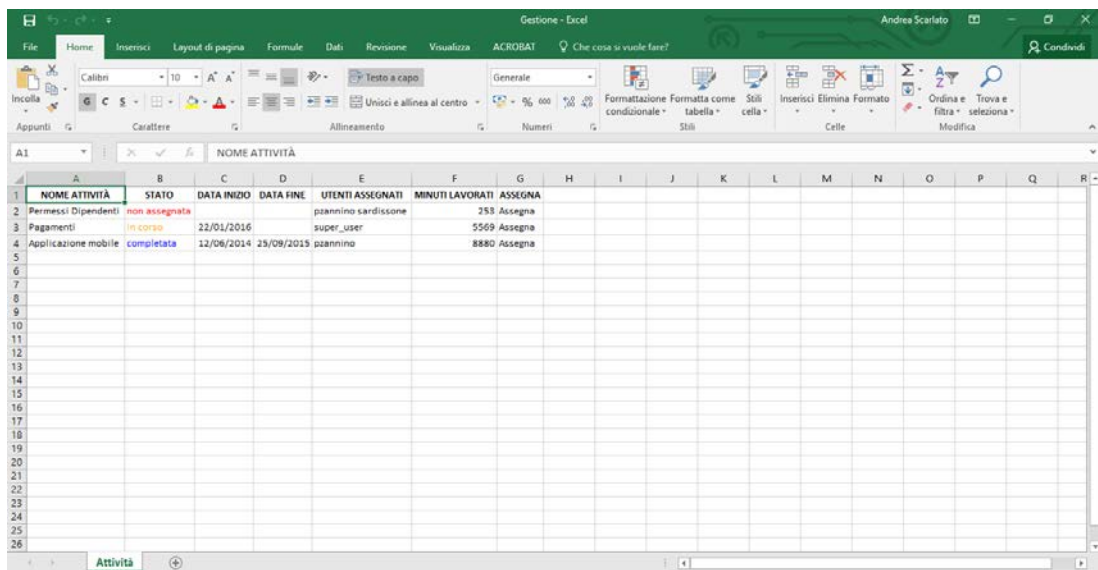
5.5 Esportazione tabelle in excel e pdf

Sempre più spesso sorge la necessità di permettere ad un utente di esportare il contenuto di una pagina in un formato più gestibile e adeguatamente formattato. Pensiamo ad esempio ad una Web Application che mostra a video tabelle o grafici. L'obiettivo è far sì che queste siano accessibili attraverso un documento di Excel o di un file pdf. In questo modo è possibile visualizzare i risultati in un file PDF in Adobe Reader, o di manipolare i dati esportati in un foglio di calcolo di per l'utilizzo offline. È possibile esportare un documento nei formati seguenti:

File Pdf: consente di esportare il documento in un file PDF utilizzando un lettore, ad esempio Adobe Reader. È possibile visualizzare il PDF su qualsiasi computer.



Foglio di calcolo Excel (xls): consente di esportare il documento in un foglio di calcolo di Microsoft Excel per ulteriori modifiche e utilizzi.



6 Conclusioni

Nel corso dei vari capitoli, ho illustrato le varie tecnologie che si utilizzano nell'ambito dello sviluppo delle applicazioni. Successivamente ho cercato di spiegare e far comprendere i motivi principali che mi hanno spinto a studiare e a sviluppare le applicazioni cross-platform con le tecnologie di cui mi sono avvalso. Infine mi sono concentrato ad esporre quella che era la progettazione e l'implementazione che ho svolto durante il periodo di realizzazione dell'applicazione.

Il modello finale rispetta ogni caratteristica e funzionalità progettata. Sicuramente l'applicazione è migliorabile sotto tanti aspetti, inizialmente era prevista l'acquisizione delle coordinate GPS, ma per motivi di tempo non è stato possibile integrare questa parte. In ogni caso l'obiettivo principale era quello di apprendere le tecnologie per lo sviluppo cross-platform.

Il framework Ionic ha soddisfatto appieno i requisiti e le aspettative, sono riuscito a realizzare un'applicazione funzionante sui principali sistemi operativi in pochissimo tempo e con l'utilizzo delle tecnologie web. Le applicazioni ibride sono veramente facile da personalizzare, ma cercare di creare un look and feel nativo può essere impegnativo. Tutto sommato, le applicazioni ibride sono ancora solo un po' indietro in termini di prestazioni, ma Ionic certamente ha fatto un lavoro fantastico riducendo il più possibile il gap con le applicazioni native.

Guardando un futuro non troppo lontano, sono convinto che lo sviluppo delle applicazioni mobile possa seriamente orientarsi prettamente verso la programmazione cross-platform, sfavorendo quella nativa.

7 Ringraziamenti

Il mio cammino universitario è stato un percorso pieno di ostacoli e molto impegnativo, questi anni mi hanno insegnato molto a livello didattico ma soprattutto a livello umano. Indubbiamente la prima lezione è che nella vita bisogna credere in sé stessi, inseguendo quelle che sono le proprie ambizioni e le proprie passioni. Appunto quelli che sono i propri traguardi, ma non solo quelli personali ma anche quelli collettivi, delle persone che ci stanno intorno. Il mio primo ringraziamento va al Signore, perché nei momenti difficili ho sentito la sua presenza al mio fianco. Ringrazio i miei genitori, mio padre che è sicuramente un esempio di grande onestà e altruismo da cui prendere esempio e quindi anche un riferimento per me, e mia madre. Sei una persona davvero speciale, non ti ringrazio abbastanza per tutto quello che sei e che fai, colgo questo momento per dimostrarti il mio affetto. Sei stata una roccia dove aggrapparmi nei momenti dove sentivo il vuoto sotto di me, mi sei stata vicina nei momenti difficili, non solo del mio percorso universitario ma anche nella quotidianità. Ringrazio i miei fratelli, Francesco per avermi fatto sorridere nei momenti che ne avevo bisogno, e Andrea, per avermi dato il supporto morale che mi serviva quando mi avvicinavo a questo traguardo e sentivo la pressione crescere. Un particolare ringraziamento va a due persone davvero straordinarie. mio nonno Roy e il mio grande amico Desmond. Probabilmente non avrei raggiunto questo importante traguardo senza il loro aiuto, sia economico ma soprattutto morale. Persone di grande altruismo e di cuore grande, siete degli esempi importanti. Ringrazio la comunità Bethel e tutte le persone che mi sono state vicine. Ringrazio tutti i miei parenti: gli zii, e i miei cugini. Grazie a tutti gli amici che mi hanno supportato ma soprattutto supportato in questi anni. Ringrazio l'azienda e miei tutors per avermi seguito in questa esperienza formativa. Infine ringrazio il prof. D'Ambrosio e il prof. Spataro, per la loro professionalità ma soprattutto per la grande umanità che mi hanno dimostrato.

Grazie a tutti voi,

Alessandro