

Giacomo Quadrio

*Drawing App*

## Resoconto stage

### Sommario

Il presente documento riassume le attività svolte durante la realizzazione del progetto di stage Drawing App, ponendo particolare attenzione sugli standard studiati, sulle tecnologie implementate e sul comportamento dei dispositivi mobile.

### Informazioni documento

**Nome:** Resoconto\_stage\_V1.0.pdf

**Versione:** 1.0

**Data creazione:** 25/07/2012

**Data ultima modifica:** 27/07/2012

**Stato del documento:** Approvato

## Registro delle modifiche:

Versione	Data	Modifiche effettuate
1.0	27/07/2012	Inserito capitolo 6 - Conclusioni
0.5	27/07/2012	Inseriti capitoli 4 e 5
0.4	26/07/2012	Inserito capitolo 3 – Fase di analisi
0.3	26/07/2012	Inseriti capitoli 2 – Strumenti e risorse
0.2	25/07/2012	Inseriti capitolo 1 - introduzione
0.1	25/07/2012	Impostata struttura generale

Tabella 1: registro delle modifiche

## Indice generale

1	Introduzione.....	4
1.1	Scopo del documento.....	4
1.2	Scopo del prodotto.....	4
1.3	Glossario.....	4
1.4	Riferimenti.....	4
1.4.1	Normativi.....	4
2	Strumenti e risorse.....	5
2.1	Hardware.....	5
2.2	Editor e versionamento.....	5
2.3	Strumenti utilizzati.....	5
3	Fase di analisi.....	6
3.1	Concept dell'applicazione.....	6
3.2	Studio delle tecnologie.....	6
3.2.1	Tecnologie per il disegno.....	6
3.2.2	Curve di Beziér .....	7
3.2.3	Tecnologie per la registrazione audio.....	7
3.2.4	Tecnologie per la condivisione del foglio di lavoro.....	8
3.2.5	Considerazioni finali.....	8
4	Realizzazione dell'applicativo.....	10
4.1	Componente di disegno.....	10
4.2	Componente di registrazione e riproduzione audio.....	10
4.2.1	Registrazione audio.....	10
4.2.2	Riproduzione audio.....	11
5	Compatibilità con i dispositivi.....	12
5.1	LG P500 Optimus One.....	12
5.2	Acer Iconia A500.....	12
6	Conclusioni.....	13
6.1	Rapporto preventivo/consuntivo.....	13
6.2	Conoscenze possedute ed acquisite.....	15

# 1 Introduzione

## 1.1 Scopo del documento

Lo scopo del documento è quello di esporre le varie attività svolte durante la realizzazione del progetto di stage Drawing App, sia dal punto di vista dello sviluppo software che di studio e ricerca tecnologica.

## 1.2 Scopo del prodotto

Il prodotto ha lo scopo di realizzare un'applicazione dedicata a dispositivi mobile come tablet e smartphone, con lo scopo di permettere all'utente di prendere appunti con varie modalità. L'applicazione prevede quindi che siano implementate varie funzionalità tipiche dei programmi dedicati al disegno o alla stesura di appunti ed anche altre funzionalità più particolari come la possibilità di registrare l'audio ambientale e sincronizzarlo con il testo o tratti presi.

## 1.3 Glossario

Per praticità e a scanso di equivoci, verrà creato un apposito documento denominato "Glossario" che conterrà tutti i termini e le sigle utilizzate nel presente documento. Questi termini verranno contrassegnati all'interno del testo con un'apposita sottolineatura.

## 1.4 Riferimenti

### 1.4.1 Normativi

- Documento di Analisi dei Requisiti, v 1.0 (Analisi\_dei\_requisiti\_V1.0.pdf)
- Documento di Specifica tecnica, v 1.0 (Specifica\_tecnica\_V1.0.pdf)

## **2 Strumenti e risorse**

Di seguito verranno elencati gli strumenti e le risorse, software ed hardware, utilizzate all'interno del progetto Drawing App. Per maggiori informazioni consultare il documento di Specifica tecnica, capitolo 2.

### **2.1 Hardware**

- Persona Computer con S.O. Mac OSX Lion 10.7.4
- Spart Phone LG Optimus One P500
- Tablet Acer Iconia A500

### **2.2 Editor e versionamento**

- Eclipse 3.7.2 (per la creazione ed i test dell'applicativo Android)
- Sublime Text 2 (per la realizzazione del codice Javascript, [HTML5](#) e [CSS](#))
- Git (strumento per il versionamento)

### **2.3 Strumenti utilizzati**

- HTML5
- Javascript
- JQuery 1.7.2
- JSON
- Android SDK
- PhoneGap – Cordova 1.8.1

## 3 Fase di analisi

In questo capitolo verrà illustrato innanzitutto il concept iniziale proposto dall'azienda Zucchetti S.p.a. relativo all'applicazione Drawing App. Verranno poi illustrate le successive fasi di analisi svolte per verificarne la fattibilità e lo studio delle tecnologie, con le annesse conclusioni raggiunte.

### 3.1 Concept dell'applicazione

Il concept di partenza dell'applicazione prevedeva che si studiasse la fattibilità di un'applicazione Web per la stesura di appunti, dedicata a dispositivi mobile quali tablet e smartphone. L'applicazione non avrebbe dovuto richiedere quindi d'essere installata, né quindi l'utilizzo di particolari plug-in. L'unico pre-requisito di cui l'utente doveva essere in possesso era un browser web compatibile con HTML5 e JavaScript.

L'applicazione era volta ad offrire uno strumento multimediale per la stesura di appunti con notevoli funzionalità a disposizione: tra le quelle che dovevano essere esplorate vi era la correzione del tratto tramite curve di Bezier, ma soprattutto la possibilità di registrare la traccia audio proveniente dall'esterno e nel mentre prendere anche appunti. Una volta terminata la registrazione, doveva essere poi possibile selezionarla, ascoltarla e visionare in tempo reale la stesura dei tratti in sincronia con l'avanzare della traccia. Un'altra funzionalità proposta era poi quella di poter prendere appunti in maniera condivisa tra più utenti, in modo da interagire nella stessa area di lavoro.

### 3.2 Studio delle tecnologie

Prima di sviluppare l'applicazione vera e propria è stata necessaria un'accurata fase di studio delle tecnologie disponibili. Questa operazione si è rivelata fondamentale in quanto ha evidenziato alcuni limiti delle tecnologie attualmente disponibili per dispositivi mobile, determinando quindi in che direzione far evolvere il progetto di stage, quali funzionalità adottare e quali invece accantonare.

#### 3.2.1 Tecnologie per il disegno

La strada più ovvia da percorrere per poter realizzare un'applicazione di disegno per dispositivi mobile tramite web è quella di utilizzare HTML5 e Javascript. HTML5 mette infatti a disposizione il tag `<canvas>` che permette di poter disegnare forme, linee, renderizzare testo, importare immagini, manipolare pixel ed esportare il contenuto in un file statico. Tale strumento deve però essere utilizzato in concomitanza con apposite funzioni in javascript che permettono di accedere al drawing context e disegnare all'interno dell'area. La peculiarità del canvas sta nel fatto che, ogni qualvolta qualcosa viene disegnato al suo interno, questo viene trasformato in semplici pixel. Tale caratteristica, se da un lato rende canvas molto leggero da utilizzare, dall'altra rappresenta una limitazione notevole alla sua gestione: se si vogliono manipolare le informazioni che vi sono state disegnate è necessario tenere traccia, tramite apposite strutture, di tutto ciò che è stato eseguito. Se infatti vogliamo eliminare l'ultimo tratto disegnato all'interno del canvas, sarà necessario ripulire l'intera area e ridisegnare tutto ciò che è stato tracciato in precedenza, ad eccezione dell'ultimo tratto.

### 3.2.2 Curve di Beziér

Le curve di Beziér rappresentano uno strumento molto interessante per poter migliorare l'aspetto visivo del tratto tracciato. Beziér permette infatti di presentare delle linee smussate, caratteristica da tenere in considerazione vista lo scopo dell'applicazione oggetto di stage. A tal proposito è stato condotto uno studio di ricerca per la sua eventuale implementazione e sono stati realizzati dei semplici test per verificarne la fattibilità ed efficienza.

JavaScript mette a disposizione, di default, una funzione molto utile denominata "bezierCurveTo" che permette di disegnare, per l'appunto, una curva di Beziér. Il prototipo di tale funzione è il seguente:

```
bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)
```

e per poter tracciare una curva, necessita d'avere le coordinate di due punti di controllo e del punto di arrivo fino a cui disegnare la curva. Essa è in grado di disegnare curve con una certa precisione e di differenti forme.

Esiste poi anche un'altra funzione per tracciare curve il cui prototipo è:

```
cubicCurveTo(cpx, cpy, x, y)
```

ed essa necessita invece di un unico punto di controllo e di un punto d'arrivo. Ciò però comporta che la curva che è possibile tracciare possieda una minor precisione e le forme che possono essere disegnate sono più limitate.

L'implementazione del disegno del tratto tramite curve di Beziér richiede quindi che, dall'array di punti che vengono registrati al verificarsi dell'evento touch, siano individuati i punti di controllo tra i vari segmenti di cui è costituito un tratto. Una volta individuati, si procederà quindi alla stesura vera e propria.

I test condotti sulla versione preliminare della correzione del tratto hanno portato ad alcuni risultati e considerazioni interessanti: tale funzionalità, se da un lato permette d'avere dei tratti più armoniosi e visivamente migliori, dall'altro aumenta non poco la pesantezza dell'applicazione sui dispositivi tablet, causando un certo ritardo nella correzione del tratto disegnato e, a volte, anche del tracciamento del tratto stesso. Tale problema è dovuto alla natura del canvas: come evidenziato nella sezione 3.2.1, ogni qualvolta sia necessario modificare il contenuto del canvas si dovrà anche procedere alla pulizia dell'area e alla ristesura dell'intero disegno. Sorge quindi una domanda ovvero se ha davvero senso sacrificare la velocità e reattività dell'applicazione in favore di un aspetto più accattivante.

### 3.2.3 Tecnologie per la registrazione audio

Le ricerche relative alla registrazione delle tracce audio via browser ha evidenziato come le tecnologie attualmente a disposizione siano ancora molto acerbe. Attualmente, il sistema più funzionale ed efficiente per realizzare ciò su computer desktop è tramite l'ausilio di plug-in in flash player, grazie ai quali è possibile accedere alle risorse hardware della macchina come web-cam o microfono.

HTML5 attualmente mette a disposizione funzionalità che permettono di ottenere il

controllo di alcune componenti hardware del computer come microfono o web-cam. Ciò è realizzato tramite delle API, denominate Web RTC, che attraverso la funzione `getUserMedia()`, consentono di ottenere gli input desiderati. Attualmente le API di HTML non sono però in grado di consentire la registrazione degli streaming video ed audio in input al device. Inoltre, attualmente solo un limitato numero di browser web desktop è in grado di supportarle, tra cui le versioni beta di Google Chrome denominata Canari ed opera Next. Per quanto riguarda i dispositivi mobile, anche sotto questo versante il supporto alle API di HTML5 e alla funzione `getUserMedia()` è molto limitato ed attualmente presente unicamente nella versione beta di Opera Mobile e Chrome per dispositivi Android.

Al momento non è quindi possibile effettuare la registrazione dei media provenienti da webcam o microfoni dei dispositivi desktop e mobile, a meno che non si utilizzino plug-in o applicazioni native.

### 3.2.4 Tecnologie per la condivisione del foglio di lavoro

L'ultimo punto su cui il concept iniziale si basava era legato alla condivisione, da parte di più utenti, del medesimo foglio di lavoro. Anche in questo caso è stata realizzata una versione preliminare di prova per verificarne la fattibilità su dispositivi mobile ed è stato utilizzata la libreria javascript denominate JWebSocket. Queste librerie permettono di condividere un'area di lavoro comune all'interno della quale, in questo caso, è possibile disegnare in maniera simultanea sullo stesso canvas. Per quanto riguarda la compatibilità, JWebSocket è compatibile con le ultime versioni dei maggiori browser web desktop, mentre il supporto ai dispositivi mobile è totalmente assente, a parte nell'ultima versione del browser Chrome.

### 3.2.5 Considerazioni finali

La fase di studio tecnologico ha portato a delle importanti conclusioni: non solo è stato possibile definire quali strumenti adottare per svolgere determinate funzioni, ma sono state anche definite quali funzionalità fosse possibile realizzare e quali invece no.

Per realizzare la base dell'applicativo si è scelto di utilizzare quindi il tag `<canvas>` di HTML5 in combinazione con Javascript in quanto unico mezzo con cui realizzare, tramite web, ciò che il progetto di stage richiedeva.

La realizzazione della correzione del tratto tramite curve di Beziér è stata invece scartata in quanto, nonostante i vantaggi estetici che avrebbe comportato, non riusciva a garantire una velocità della stesura del tratto soddisfacente. I test sono stati condotti su più dispositivi ed il risultato ottenuto, nonostante alcune piccole variazioni, è stato il medesimo. Si è in conclusione scelto di dar priorità all'efficienza piuttosto che all'estetica: un'applicazione per il disegno che non è in grado di rispondere adeguatamente agli input dell'utente non è decisamente consigliata.

L'esplorazione delle tecnologie relative alla registrazione audio ha dato esito negativo, evidenziando una situazione decisamente acerba degli strumenti a disposizione. Gli unici strumenti disponibili per effettuare la registrazione dell'audio tramite browser web desktop sono rappresentati da plug-in in flash o altri linguaggi, mentre su dispositivi mobile, attualmente, non esiste nulla che permette di fare ciò. Poiché la registrazione dell'audio rappresenta uno dei punti focali su cui verte il progetto di stage, si è deciso di implementare tale funzionalità sui dispositivi mobile tramite l'utilizzo di un Framework mobile di sviluppo open source denominato PhoneGap. Questo strumento permette di realizzare del codice HTML, CSS e Javascript, ed importarlo all'interno di



un'applicazione nativa per smartphone, dando la possibilità di accedere all'hardware come memoria interna, microfono, fotocamera e via dicendo. In questo modo si manterrà in parte la natura web dell'applicazione, garantendo comunque che una delle funzionalità fondamentali venga realizzata. Nel caso di Drawing App si è scelto di sviluppare l'applicazione per dispositivi Android in quanto non comporta costi aggiuntivi di licenza ed è possibile provare l'applicazione direttamente sui dispositivi, senza quindi utilizzare emulatori.

Vista infine la difficoltà nel reperire tecnologie per la condivisione del foglio di lavoro compatibili con i dispositivi mobile e, soprattutto, la decisione di realizzare un'applicazione nativa locale per smartphone, si è convenuto di accantonare tale funzionalità.

Al termine di questa fase di analisi tecnologica è emerso quindi un nuovo concept dal quale si è partiti per dare vita all'applicazione oggetto di stage, denominata Drawing App.

## 4 Realizzazione dell'applicativo

Verrà illustrato di seguito come sono state realizzate le funzionalità principali del software riguardanti la stesura del tratto all'interno del canvas e la registrazione e riproduzione di una traccia audio. Per maggiori informazioni sulle classi ed i metodi implementati consultare il documento di Specifica tecnica.

### 4.1 Componente di disegno

Come già accennato, per realizzare l'area di disegno si è fatto uso del linguaggio HTML5 e Javascript, in particolare verrà utilizzato il tag `<canvas>` per offrire lo spazio su cui compiere i tratti. Per poter compiere però queste operazioni, dovremo servirci di alcune funzioni che Javascript mette a disposizione per poter intercettare gli input di tipo touch.

Attraverso la funzione `getElementById` richiamata sull'oggetto `document`, che rappresenta la pagina HTML, si sono ottenuti i vari componenti necessari all'interazione per il disegno. Innanzitutto si è preso il `div` adibito a contenere l'area di disegno e vi si è appeso il nuovo oggetto appositamente creato di tipo canvas. Sono stati poi estratti i vari pulsanti adibiti alle operazioni descritte nel documento di Analisi dei requisiti.

Per permettere però l'effettiva stesura del tratto, è necessario appendere all'area di disegno degli appositi eventi che vanno a richiamare quelle funzioni di Javascript quando avviene una precisa interazione con lo schermo tramite tocco. Nel dettaglio le funzioni relative agli eventi touch sono:

- **touchstart**: evento che si attiva quando un dito viene poggiato sulla superficie dello schermo. E' possibile determinare la sua posizione in termini di coordinate x, y.
- **touchmove**: si attiva quando un dito già poggiato sulla superficie dello schermo si sposta. Esso traccia i movimenti, fornendo anche in questo caso le coordinate x, y del dito.

Ogni qualvolta questa tipologia di eventi si verifica, vengono chiamate delle funzioni che andranno ad eseguire operazioni sull'oggetto canvas. La prima, *touchstart*, richiamerà la funzione *preDraw* che effettuerà le operazioni di preparazione per la stesura del tratto. La seconda, *touchmove*, si occuperà invece di richiamare la funzione *draw* che traccia effettivamente la linea. Entrambe queste operazioni, oltre che a disegnare il tratto, si occuperanno anche di definire gli elementi di un apposito array con le informazioni del tratto che si sta tracciando in un preciso istante. Le informazioni riguardano l'id univoco del tratto, i punti di cui è composto, lo spessore, il colore, la data in cui sono stati stesi ed altre informazioni legate alle tracce audio in registrazione. L'utilizzo di un array per tenere traccia di tutto ciò è quindi fondamentale per tutte quelle operazioni che andranno poi a modificare l'area di disegno in quanto, una volta che il tratto è stato eseguito, questo si trasformerà in pixel all'interno del canvas e le informazioni ad esso legate non potranno più essere reperibili.

### 4.2 Componente di registrazione e riproduzione audio

#### 4.2.1 Registrazione audio

Javascript, così come HTML5 non mettono purtroppo a disposizione strumenti per effettuare l'accesso all'hardware del dispositivo mobile e registrare tracce audio. Ci si è quindi affidati al framework PhoneGap in abbinamento con l'SDK Android per

effettuare ciò.

Come per i pulsanti dedicati alla modifica del tratto, anche per quelli legati alla registrazione ci si è affidati agli *addEventListener* per rilevare la pressione su di essi ed avviare le apposite funzioni. La funzione *recordAudioStart*, che possiede infatti un ascoltatore di tipo *mousedown*, avvia la registrazione di una nuova traccia. Quando avviene ciò, non solo si creerà un nuovo file audio che verrà poi immagazzinato all'interno della memoria del dispositivo, ma verrà anche definito l'apposito array contenente tutte le informazioni delle tracce audio registrate in una certa sessione. Le informazioni sono l'id della traccia audio, il nome del file, la data di avvio registrazione, la data di termine registrazione e la pagina di appartenenza. Una volta avviata la registrazione sarà poi definita anche una variabile che si occuperà di indicare, al momento della stesura di un tratto, a quale traccia audio esso appartiene. Questo array è necessario poiché sarà utilizzato per definire la lista dei media disponibile in una certa pagina, nonché per poter accedere ad una certa traccia audio nella memoria del dispositivo e riprodurla.

La funzione di *recordAudioStop* si occuperà di fermare la registrazione della traccia audio, ridefinire il menù delle tracce audio disponibili in una certa pagina con la traccia appena registrata ed infine ripotrerà a “undefined” la variabile che definisce quale traccia audio è attualmente in registrazione

#### 4.2.2 Riproduzione audio

Anche per la riproduzione audio di file contenuti all'interno della memoria del dispositivo, ci si è ancora una volta affidati al framework PhoneGap in abbinamento con l'SDK Android per poter eseguire tali operazioni.

Dopo aver effettuato la registrazione audio, avremo tutte le informazioni per poter avviare la riproduzione sincronizzata della traccia audio e dei tratti disegnati durante la sua registrazione. Avviata la funzione *playAudioDraw*, andremo innanzitutto a cercare la traccia audio che vogliamo riprodurre all'interno dell'apposito array. Una volta trovata, si definirà il tempo totale della traccia audio e quali linee vi appartengono. Effettuato ciò, si coloreranno in grigio i tratti che dovranno essere ridisegnati e si avvierà la riproduzione del file MP3. L'ultima operazione eseguita è l'invocazione della funzione *sincroDraw* che disegnerà in maniera sincronizzata il tratto. La funzione è una funzione ricorsiva temporizzata tramite il metodo *setTimeout*, che attenderà un certo tot di tempo prima di tracciare la linea tra un punto ed un altro. Questo tempo è calcolato tenendo in considerazione quanti millisecondi sono trascorsi tra la stesura di un punto ed un altro.

## 5 Compatibilità con i dispositivi

I dispositivi verso cui è indirizzata l'applicazione sono principalmente smartphone e tablet dotati di schermo touch screen e sistema operativo Android. Come già accennato, i dispositivi scelti per effettuare le prove sono stati uno smartphone LG P500 Optimus One ed un tablet Acer Iconia A500, entrambi dotati di sistema operativo Android 4.0.4 Ice Cream Sandwich.

### 5.1 LG P500 Optimus One

L'applicazione, una volta installata, impiega un certo quantitativo di tempo per essere avviata, questo a causa probabilmente dell'hardware non esattamente performante e recente. Avviata l'operazione di disegno si noterà immediatamente quanto questa sia lenta e poco performante: vari test condotti anche con altre applicazioni che utilizzano una tecnologia simile, nonché smartphone, hanno infatti evidenziato che, su dispositivi datati e non molto recenti, il rendering del canvas via javascript risulta essere molto lento tale da rendere l'applicazione quasi inutilizzabile.

Le altre funzionalità dell'applicazione sono invece perfettamente funzionanti e compatibili in maniera completa ed efficiente: la funzione di registrazione delle tracce audio avviene in maniera corretta, così come il caricamento delle immagini, il salvataggio dei file ed il loro ripristino.

### 5.2 Acer Iconia A500

L'applicazione, una volta installata, a differenza dell'LG P500 impiega pochissimo tempo per essere avviata. Avviata l'operazione di disegno, il tratto verrà tracciato in maniera efficiente e reattiva, seguendo di pari passo il movimento delle dita dell'utilizzatore. I risultati decisamente migliori rispetto a quelli ottenuti dallo smartphone sono dovuti all'hardware più performante ed aggiornato.

Le altre funzioni dell'applicazione sono perfettamente funzionanti e compatibili in maniera completa ed efficiente: la funzione di registrazione delle tracce audio avviene in maniera corretta, così come il caricamento delle immagini, il salvataggio dei file ed il loro ripristino.

## 6 Conclusioni

### 6.1 Rapporto preventivo/consuntivo

Di seguito viene riportata la tabella riassuntiva delle ore che sono state dedicate alle varie attività, mettendole a confronto tra quelle pianificate e quelle effettivamente svolte.

Attività	Preventivo	Consuntivo	Scostamento
Formazione/analisi	30	50	+20
Progettazione	40	80	+40
Codifica	160	120	-40
Verifica	50	40	-10
Documentazione	40	30	-10
<b>TOTALE</b>	<b>320</b>	<b>320</b>	<b>0</b>

Tabella 2: confronto ore preventivo / consuntivo

A causa della natura molto sperimentale del progetto di stage, le ore preventivate hanno subito alcune variazioni e la distribuzione tra le varie attività è cambiata. Di seguito verranno ora mostrati alcuni grafici relativi a ciò.

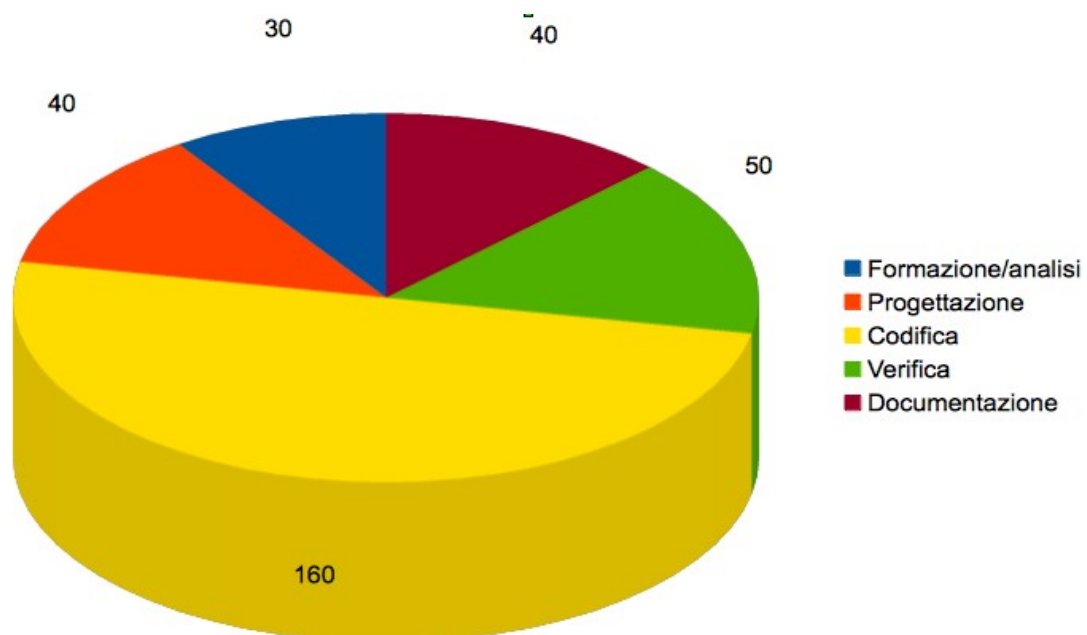


Figura 1: ore a preventivo

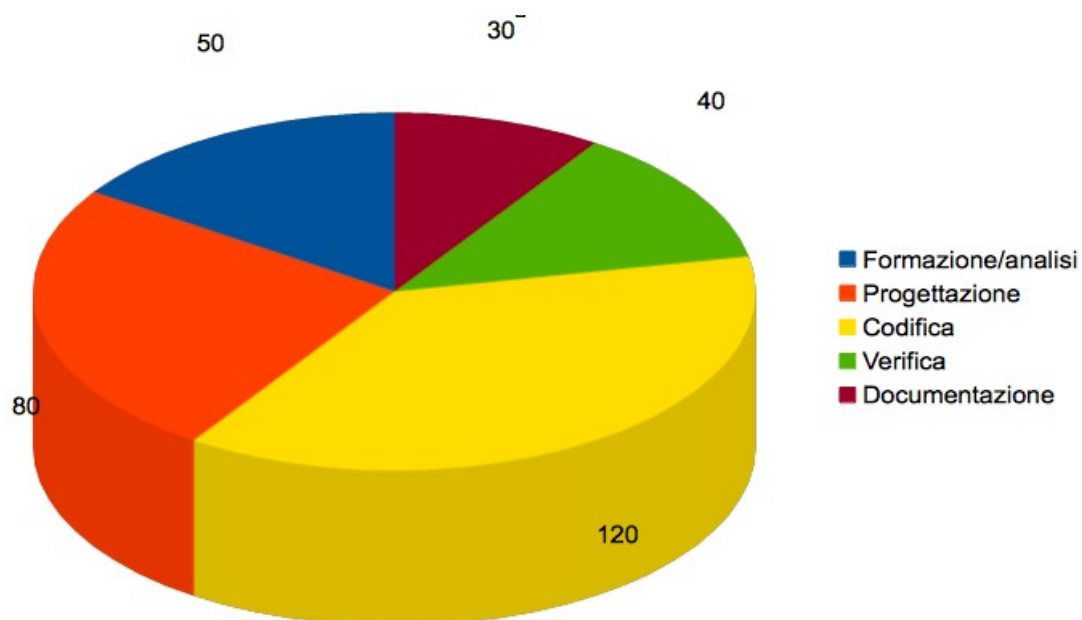


Figura 2: ore a consuntivo

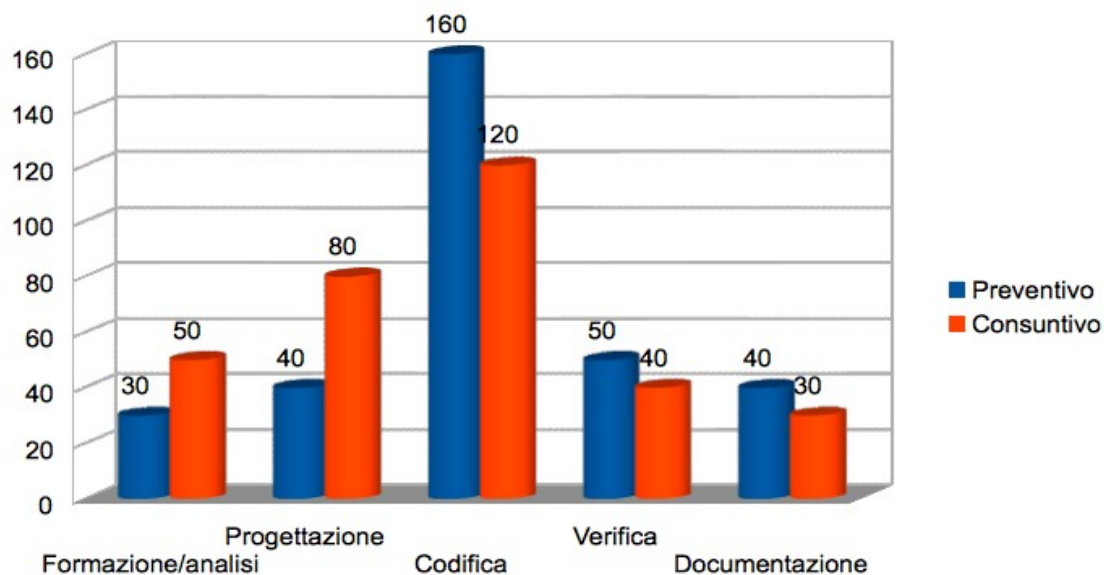


Figura 3: confronto ore preventivo/consuntivo

Come si può notare dalla tabella riassuntiva degli orari e anche dall'istogramma di confronto tra le ore preventivate e consuntivate, c'è un certo scostamento tra tutte le attività. Quelle che hanno subito un differenziale minimo sono state le attività di verifica e stesura dei documenti, dove la variazione è stata di circa 10 ore, non allontanandosi quindi molto da quando inizialmente previsto.

Le attività che invece hanno subito i maggiori cambiamenti in fatto di distribuzione oraria sono le seguenti:

- **Formazione / analisi:** la fase ha richiesto una maggiore quantità di ore in quanto è stato necessario dedicare più tempo allo studio e ricerca tecnologici poiché le tecnologie coinvolte nel progetto si sono rivelate essere molto nuove ed acerbe e quindi non completamente conosciute.

Come descritto nel capitolo 3.2, si sono innanzitutto analizzate tutte le funzionalità cruciali dell'applicazione e si è svolta un'accurata ricerca su quali tecnologie potessero essere utili per realizzarle. In particolare, la decisione di esplorare l'alternativa dell'applicazione nativa ha comportato lo studio di elementi non inizialmente considerati come il framework PhoneGap e l'Android SDK.

- **Progettazione:** la fase ha visto duplicare il quantitativo orario ad essa dedicato, passando da 40 ore ad 80. Questo è stato necessario in quanto, come già accennato, l'applicazione si è dovuta spostare da un'ambiente prettamente web ad uno ibrido. L'introduzione quindi del framework PhoneGap e dell'ambiente Android ha necessitato un'accurata fase di ulteriore progettazione ed adattamento dell'idea iniziale dell'applicazione, con conseguente ideazione di un nuovo prototipo. Parte del tempo in più speso è stato dovuto in parte anche al fatto che le tecnologie utilizzate nel progetto non erano completamente conosciute da me e quindi è stato necessario più tempo per poter ideare una progettazione adeguata.
- **Codifica:** il quantitativo orario della codifica è invece diminuito di ben 40 ore rispetto a quanto ideato inizialmente. Ciò è stato possibile grazie ad un'accurata e dettagliata fase di progettazione che ha portato quindi a risparmiare un notevole monte ore, impiegato e ridistribuito poi in altre attività.

## 6.2 Conoscenze possedute ed acquisite

Le conoscenze da me possedute ad inizio stage erano di tipo basilare e riguardavano principalmente le tecnologie più comuni utili al progetto. Esse riguardavano principalmente:

- Linguaggi HTML5 e CSS
- Ambiente di sviluppo Android SDK

Le fasi di formazione, analisi e progettazione sono quindi state fondamentali per consentire la crescita del bagaglio personale delle conoscenze in termini di tecnologie e poter quindi acquisire i giusti strumenti per portare a compimento gli obiettivi prefissati del progetto di stage.

Nel dettaglio ho esplorato in maniera notevole il linguaggio Javascript, vero e proprio cuore dell'applicazione Drawing App, arrivando ad un livello di conoscenza

personalmente soddisfacente. Il progetto di stage mi ha poi portato ad apprendere inoltre l'utilizzo del tag `<canvas>` di HTML5, che si è rivelato essere molto interessante ed al contempo anche molto complesso da gestire se lo si vuole utilizzare per un'applicazione di disegno. Ho poi trovato molto stimolante l'utilizzo del framework PhoneGap, uno strumento molto potente ed in grado di offrire al programmatore molti strumenti utili una volta che se ne è appreso il funzionamento.

Uno degli elementi che mi ha infine molto soddisfatto è stato sicuramente lo sviluppare un'applicazione in ambiente mobile. Esso è un campo ancora molto giovane ed ho riscontrato, più di qualche volta alcune, difficoltà soprattutto nella gestione degli eventi via touch screen. La conoscenza maggiore che però ho acquisito è senza ombra di dubbio proprio l'interazione, tramite HTML5 e Javascript, con il touch screen: la gestione degli eventi ad esso legati è davvero interessante e ho appreso quanto sia importante una corretta e non approssimativa gestione di questi, soprattutto in un ambito come quello di un'applicazione per il disegno, dove la precisione del tocco è fondamentale.