

Università Ca' Foscari di Venezia

Dipartimento di Scienze Ambientali, Informatica e Statistica

Corso di Laurea in Informatica

**Corso di Ingegneria del Software AA 2017/2018**



# Piano di Progetto

**Gruppo SATA:**

Andrea Wohlers (854452)  
Susanna Lorenzini (854153)  
Alessandro Busatto (856329)  
Tommaso Michieletto (834470)

## Il team SATA



Susanna Lorenzini



Andrea Wohlers



Tommaso Michieletto



Alessandro Busatto

# Dettagli Del Documento

---

**Tipo Documento:** Piano di Progetto

**Data Prima pubblicazione:** 23/10/2017

**Redatto e Revisionato da:**

Andrea Wohlers  
Susanna Lorenzini  
Alessandro Busatto  
Tommaso Michieletto

**Allegati:**

1. Diagramma Gantt
2. Diagramma Pert v1
3. Diagramma Pert v2

Versioni successive e modifiche al documento:

Versione	Data Rilascio	Modifiche apportate
1.0	23/10/2017	-
2.0	31/01/2018	Tabella Attività e Funzioni, Gantt, Pert

# Indice

<b>Sezione 1 - Introduzione</b>	<b>6</b>
1.1 Overview del Progetto	6
1.2 Deliverables del Progetto	6
1.3 Evoluzione del Progetto	7
1.4 Materiale di Riferimento	7
1.5 Definizioni e Abbreviazioni	7
<b>Sezione 2 - Organizzazione del Processo</b>	<b>9</b>
2.1 Modello del Processo	9
2.2 Struttura Organizzativa	9
2.3 Interfacce Organizzative	10
2.4 Responsabilità di Progetto	10
<b>Sezione 3 - Processi Gestionali</b>	<b>11</b>
3.1 Obiettivi e Priorità	11
3.2 Assunzioni, Dipendenze, Vincoli	11
3.3 Gestione dei Rischi	12
3.4 Meccanismi di Monitoraggio e di Controllo	14
3.5 Pianificazione dello Staff	14
<b>Sezione 4 - Descrizione dei Processi Tecnici</b>	<b>15</b>
4.1 Metodi, Strumenti e Tecniche	15
4.2 Documentazione del Software	15
4.3 Funzionalità di Supporto al Progetto	15
<b>Sezione 5 - Pianificazione del Lavoro, delle Risorse Umane e del Budget</b>	<b>16</b>
5.1 Work Breakdown Structure	16
5.1.1 WBS	16
5.1.2 Attività e Funzioni	18
5.1.3 Gantt	19
5.1.4 Pert	20
5.2 Dipendenze	22
5.3 Risorse Necessarie	22

5.4 Allocazione del Budget e delle Risorse	22
5.5 Pianificazione	22

# Sezione 1 - Introduzione

## 1.1 Overview del Progetto

Lo scopo del progetto è quello di sviluppare un'applicazione per dispositivi mobili in ambiente Android che ha come target lo studente universitario e che fornisca all'utente una visione di quanto denaro viene utilizzato nei vari settori dell'università rispetto all'importo delle tasse universitarie. Questi dati verranno raccolti utilizzando gli open data forniti da Ca' Foscari relativi al bilancio.

L'applicazione è rivolta a tutti, studenti di Ca' Foscari in primis, ma anche a curiosi che vogliono informarsi su come Ca' Foscari utilizza il ricavato delle tasse annuali versate dagli studenti.

## 1.2 Deliverables del Progetto

Gli elementi che verranno consegnati si possono raggruppare in due tipologie: documentazione, cioè i Deliverables dall'1 al 4, e applicazione Android, il Deliverable numero 5.

Nella tabella seguente vengono definite le tipologie di ogni Deliverable e le rispettive scadenze:

	Tipologia di Documento	Scadenza
1	Piano di Progetto	23/10/2017
2	Documento di Analisi e Specifica	09/11/2017
3	Piano di Testing	23/11/2017
4	Documentazione di Progettazione	15/12/2017
5	Realizzazione e messa in linea	31/01/2018

I documenti verranno successivamente modificati e migliorati a seconda dell'evoluzione del progetto e della fattibilità di ogni parte dello stesso; tali modifiche andranno a formare le versioni definitive dei documenti. I documenti verranno inseriti su Git.

### 1.3 Evoluzione del Progetto

Una prima implementazione prevede la raccolta e l'analisi dei bilanci dell'Università Ca' Foscari.

Una possibile evoluzione dell'applicazione è quella dell'inserimento di più università, con l'obiettivo ultimo di inserire tutte le 68 università italiane, cosicché sia possibile non solamente visionare ed analizzare i dati relativi a Ca' Foscari, ma anche operare un confronto tra i diversi atenei.

Questo obiettivo sarà raggiungibile se le varie Università forniranno gli open data relativi ai loro bilanci.

### 1.4 Materiale di Riferimento

Per l'organizzazione del lavoro e la redazione dei documenti che verranno consegnati, sarà presa in considerazione la seguente documentazione:

- Slides + Appunti del corso di Ingegneria del Software 2017/2018 (Prof. Cortesi)
- Materiale e documentazione degli anni precedenti
- Libro di testo : "Android 5 Programming by Example" di Kyle Mew, Packt
- Libro di testo : "Android Programming for Beginners" di John Horton, Packt
- Slides + Appunti del corso di Project Management 2016/2017 (Prof. D'Orsi)

### 1.5 Definizioni e Abbreviazioni

All'interno del documento saranno utilizzate le seguenti abbreviazioni e definizioni:

- **API:** Application Program Interface, insieme delle procedure disponibili al programmatore, raggruppate a formare un set di strumenti specifici per svolgere un certo compito. Con questo termine indichiamo le librerie software disponibili.
- **App:** applicazione software dedicata ai dispositivi mobili con sistema operativo Android.
- **Bilancio:** il complesso di dati e conteggi relativi ai movimenti economici di un'amministrazione pubblica o privata.
- **Committente:** utente che commissiona il progetto.
- **Deliverable:** letteralmente, "oggetti da consegnare"; sono un prodotto, un servizio o un risultato del progetto con data e luogo di consegna, quasi sempre parziale, che può essere univocamente definito, misurato e verificato.
- **Diagramma di Gantt:** diagramma a barre che rappresenta le attività in un piano cartesiano con il tempo in ascissa e la successione delle attività in ordinata, mediante barre la cui lunghezza è proporzionale alla durata. Da questo diagramma si ricava il tempo di ogni singola attività, quando è collocata nel tempo e quanto tempo è necessario per completare l'intero progetto; tuttavia, non tiene in considerazione le interdipendenze tra le diverse attività.
- **Diagramma di Pert:** diagramma reticolare (grafo) nel quale i nodi rappresentano le attività e gli archi rappresentano i rapporti di dipendenza tra le attività. Il focus di questo diagramma è il collegamento logico tra le attività.

- **Fase di testing:** il test del software è un procedimento utilizzato per individuare carenze di correttezza, completezza ed affidabilità delle componenti software in corso di sviluppo.
- **Milestone:** Importanti traguardi intermedi nello svolgimento del progetto.
- **Open data:** dati che possono essere liberamente utilizzati, riutilizzati e distribuiti da chiunque, soggetti eventualmente alla necessità di citarne la fonte e di condividerli con la licenza con cui sono stati originariamente rilasciati.
- **Play Store:** detto anche Google Play Store, è un negozio online di applicazioni, brani musicali, pellicole cinematografiche, libri e riviste sviluppato da Google Inc. principalmente per offrire servizi ai dispositivi mobili Android.



## Sezione 2 - Organizzazione del Processo

### 2.1 Modello del Processo

Il progetto verrà sviluppato secondo il modello di tipo evolutivo: inizialmente il gruppo si atterrà alle specifiche di massima e, con l'avanzare del progetto e dello sviluppo dell'applicazione, verranno riviste le caratteristiche che porteranno ad un'eventuale modifica dei documenti.

Verranno fatti i seguenti passi fino ad arrivare alla conclusione del progetto:

- Specifica delle funzionalità
- Sviluppo del codice
- Validazione del materiale prodotto

### 2.2 Struttura Organizzativa

E' stata scelta un'organizzazione di tipo controllato decentralizzato, il team è così composto:

- ❑ Leader: **Andrea Wohlers**
- ❑ **Alessandro Busatto**
- ❑ **Tommaso Michieletto**
- ❑ **Susanna Lorenzini**

Il leader è il nostro punto di riferimento, coordina il lavoro di gruppo e sceglie dei sottogruppi per lo sviluppo del progetto.

In particolare, Alessandro e Tommaso si occuperanno della stesura del codice e della realizzazione dell'app; Andrea e Susanna si occuperanno della realizzazione e della revisione della documentazione.

Tuttavia, tutti i membri del gruppo interagiranno e lavoreranno assieme per rispettare l'obiettivo finale, cioè la consegna dell'app entro il 31/01/2018 e la redazione della documentazione completa.

I deliverables saranno consegnati tramite Git utilizzando due computer dei componenti del gruppo.

Per quanto riguarda la comunicazione abbiamo deciso di procedere con una condivisione di tutti i documenti e di tutto il codice, appoggiandoci rispettivamente a Google Drive e a GitKraken, con la possibilità in entrambi i casi di operare anche da casa su un file modificato in real time.

Settimanalmente viene indetta una riunione frontale per verificare l'andamento del progetto e per dividere i task tra i membri del gruppo.

Inoltre saranno utilizzati altri strumenti quali Skype e Google Duo, per le chiamate tra i membri del gruppo quando non è possibile incontrarsi, e Telegram, per la comunicazione diretta tra i componenti.

### **2.3 Interfacce Organizzative**

Durante lo sviluppo del progetto il gruppo si relazionerà ad entità esterne per la realizzazione e l'ottimizzazione dello stesso.

In particolare, le entità esterne di maggior rilevanza sono:

- il docente del corso professore Agostino Cortesi
- il tutor del corso Alvisè Spanò
- Tommaso Piazza (Direttore ASIT unive) con la fornitura dei dati dell'università Ca' Foscari
- possibili utenti dell'applicazione, identificati negli studenti universitari, per comprendere le richieste dell'utente target
- un campione di possibili utenti, tecnici e non, per la fase di testing e per fornire un feedback utile volto a migliorare ed ottimizzare l'applicazione

### **2.4 Responsabilità di Progetto**

Le principali funzioni ed attività da portare a termine si identificano nelle fasi di creazione e sviluppo del software, nella stesura e nella revisione continua della documentazione, fino alla realizzazione dell'applicazione finale.

Avendo scelto un'organizzazione di tipo controllato decentralizzato, il leader Andrea Wohlers coordinerà il lavoro ed assegnerà le diverse attività ai sottogruppi formati dai componenti del team, in modo che tutti lavorino in modo equo per lo svolgimento del progetto.

## Sezione 3 - Processi Gestionali

### 3.1 Obiettivi e Priorità

L'obiettivo principale è quello di portare a termine ogni Deliverable entro le date previste. La consegna dell'applicazione, completa e conforme ai requisiti, è prevista entro la data 31/01/2018 e la consegna della documentazione finale dovrà avvenire prima del colloquio frontale.

Un ulteriore obiettivo consiste nella stesura di un codice leggibile per favorire la comprensione, la revisione e la modifica dello stesso.

Le priorità identificate sono l'acquisizione degli open data necessari e delle capacità di programmazione per sistemi Android per garantire la riuscita del progetto.

### 3.2 Assunzioni, Dipendenze, Vincoli

#### Assunzioni:

L'idea del progetto parte dalla fondamentale assunzione che non solo uno studente dovrebbe essere informato su come vengono investite le proprie tasse universitarie, ma dovrebbe interessarsene egli stesso in primis. Le migliorie vengono apportate soprattutto grazie al feedback degli utenti, i quali devono avere tutte le informazioni possibili, nell'ottica della trasparenza.

Per poter usufruire dell'applicazione l'utente dovrà possedere uno smartphone con accesso a Internet e un sistema operativo Android Lollipop 5.1 o superiore.

#### Dipendenze:

Gli open data di Ca' Foscari dovranno essere il più completi e corretti possibile per poter fornire un'applicazione utile e precisa.

Inoltre, l'impegno costante dei membri del team e la partecipazione agli incontri per confrontarsi e commentare lo sviluppo del progetto sono fondamentali.

#### Vincoli:

Impegni extra-progetto dei membri, quali frequenza a lezioni universitarie e relativi appelli d'esame, sport, etc.. riducono il tempo disponibile, rendendo più difficile far coincidere gli impegni di tutti per poter lavorare insieme, pertanto le sotto-task andranno schedulate anche in base ad essi.

I membri del gruppo dovranno avere tutti i mezzi fisici e le nozioni teoriche per poter rispettare il piano di lavoro.

Infine, dovranno essere rispettate le date di consegna dei vari documenti e dell'applicazione.

### 3.3 Gestione dei Rischi

Identifichiamo i rischi con una coppia di valori: la probabilità che un rischio si verifichi e l'entità del suo impatto.

La probabilità potrà essere:

- molto bassa (MB): inferiore al 5%
- bassa (B): tra il 5% e il 24%
- media (M): tra il 25% e il 49%
- alta (A): tra il 50% e il 74%
- molto alta (MA): tra il 75% e il 99%
- certa (C): 100%

L'impatto, invece, potrà risultare:

- molto basso (MB): effetti trascurabili con conseguenze insignificanti
- basso (B): effetti limitati con conseguenze marginali
- medio (M): effetti notevoli con conseguenze potenzialmente critiche
- alto (A): effetti pesanti con conseguenze critiche
- molto alto (MA): effetti disastrosi con conseguenze catastrofiche

I rischi identificati sono i seguenti:

**R1:** stima scorretta delle tempistiche (P: media, I: alto)

**R2:** conflitti all'interno del gruppo (P: bassa, I: medio)

**R3:** mancato apprendimento del linguaggio di programmazione (P: bassa, I: molto alto)

**R4:** mancanza temporanea di uno o più membri del gruppo per problemi di salute (P: media, I: medio)

**R5:** assenza temporanea di uno o più membri per impegni universitari (P: alta, I: medio)

**R6:** abbandono di un membro del gruppo (P: bassa, I: medio)

**R7:** indisponibilità degli open data necessari (P: bassa, I: molto alto)

**R8:** perdita dei dati (P: bassa, I: molto alto)

[vedi matrice dei rischi nella pagina seguente]

		Impatto				
		Molto Basso	Basso	Medio	Alto	Molto Alto
Probabilità	Molto Alta					
	Alta			R5		
	Media			R4	R1	
	Bassa			R2, R6		R3, R7, R8
	Molto Basso					

Definiamo, inoltre, delle azioni preventive o correttive per la gestione dei rischi:

**R1:** verrà monitorato l'andamento del progetto, con particolare riferimento alle scadenze fissate, per prevedere eventuali ritardi.

**R2:** verranno mitigati e controllati eventuali conflitti e, in caso di indecisione su come procedere nella realizzazione del progetto, la decisione sarà presa dal leader.

**R3:** verranno apprese le tecniche di programmazione tramite l'utilizzo dei libri indicati nei materiali; in caso di necessità i membri del gruppo si aiuteranno a superare le lacune dei componenti in difficoltà.

**R4:** in caso di problemi di salute il lavoro degli altri membri verrà intensificato e si terrà aggiornato il componente indisposto, per permettere una ripresa rapida.

**R5:** ogni membro del gruppo comunicherà quanto prima le assenze dovute ad impegni universitari, così da permettere un'organizzazione ottimale degli altri componenti.

**R6:** i membri del gruppo revisioneranno il progetto e proporranno un piano intensivo per portarlo al termine, in modo da sopperire alla mancanza di un componente.

**R7:** se gli open data non dovessero essere disponibili il rischio dovrà essere accettato e si dovranno valutare altre proposte per lo svolgimento del progetto.

**R8:** verranno fatti dei backup frequenti e ogni membro del gruppo manterrà una copia dei materiali utili.

### 3.4 Meccanismi di Monitoraggio e di Controllo

Il gruppo si riunirà almeno una volta a settimana per monitorare l'andamento del progetto; se necessario, il numero di incontri aumenterà.

Ogni membro del gruppo potrà controllare i materiali sulla cartella condivisa su Google Drive in ogni momento e notificherà eventuali problemi nel gruppo Telegram per il confronto istantaneo tra tutti i membri.

### 3.5 Pianificazione dello Staff

Lo staff è composto da quattro membri, che si alterneranno per portare a termine tutte le fasi volte al completamento del progetto.

Le abilità precedentemente acquisite sono:

1. **Andrea Wohlers**: programmazione, conoscenze di project management.
2. **Susanna Lorenzini**: programmazione, conoscenze di project management.
3. **Alessandro Busatto**: programmazione, design, conoscenze software, conoscenze hardware.
4. **Tommaso Michieletto**: programmazione, conoscenze software, conoscenze hardware.

Le skill di base sono, inoltre, la conoscenza basilare sull'utilizzo di Android Studio, Git e lo sviluppo in Java.

Le abilità da acquisire sono la conoscenza della piattaforma Android, la programmazione Java in ambiente Android e la capacità di stendere i vari documenti in relazione al progetto. Queste conoscenze saranno acquisite tramite l'utilizzo dei materiali specificati nella sezione 1.4 Materiali di Riferimento.

## Sezione 4 - Descrizione dei Processi Tecnici

### 4.1 Metodi, Strumenti e Tecniche

I membri del gruppo utilizzeranno i loro PC per lo sviluppo del codice e la stesura dei documenti, inoltre verranno utilizzati i pc messi dei laboratori 3 e 5 del Dipartimento di Scienze Ambientali, Informatiche e Statistiche dell'Università di Ca' Foscari.

Per lo sviluppo del codice verrà utilizzato Android Studio e SDK tools basato sul software di JetBrains IntelliJ IDEA; per la fase di testing dell'app verrà utilizzato l'emulatore fornito dal suddetto tool e due smartphone forniti da due membri del gruppo, il primo è un Honor 7 sul quale gira Android 7, che in fase di sviluppo dell'app potrebbe avere anche Android 8, mentre il secondo è un Samsung Galaxy S3 sul quale gira Android 5.

Il codice verrà condiviso tra i membri del gruppo attraverso la piattaforma Git, non verrà reso pubblico agli altri utenti prima della fine del progetto.

Il linguaggio utilizzato per lo sviluppo è il Java.

La documentazione verrà condivisa utilizzando Google Drive, così da permettere una stesura interattiva e contemporanea del testo. In questo modo, i materiali saranno sempre accessibili e modificabili da ogni membro del gruppo.

### 4.2 Documentazione del Software

La versione finale del prodotto sarà accompagnata dalla seguente documentazione:

- Documento di progettazione
- Documento dei requisiti
- Documentazione relativa alle fasi di testing
- Manuale d'uso per l'utente: serve ad illustrare le funzionalità del programma
- Commenti interni al codice: utile ai membri del gruppo per capire ciò che è stato scritto da qualcun altro, faciliterà la revisione e la manutenzione del codice.

### 4.3 Funzionalità di Supporto al Progetto

Nello sviluppo del progetto si ritiene importante controllare la qualità dello stesso in ogni fase; per fare ciò, durante lo sviluppo dell'applicazione, ci saranno degli incontri pianificati per la revisione del progetto e dei documenti. In questi incontri il team si concentrerà sul verificare se il progetto rispetta le specifiche, in particolare le specifiche funzionali e tecniche.

Durante lo sviluppo dell'app verranno rilasciate delle versioni di testing ad un gruppo di utenti scelti dal team, tecnici e non, per avere dei feedback utili volti a migliorare l'app.

## Sezione 5 - Pianificazione del Lavoro, delle Risorse Umane e del Budget

### 5.1 Work Breakdown Structure

#### 5.1.1 WBS

##### 1. Specifiche di progetto

- 1.1 Definizione obiettivi generali e ambito del progetto
- 1.2 Raccolta dei requisiti generali
- 1.3 Definizione delle risorse necessarie
  - 1.3.1 Definizione delle risorse materiali richieste
  - 1.3.2 Definizione delle tipologie di skill richieste
  - 1.3.2 Definizione della composizione ottimale del team
- 1.4 Analisi di Fattibilità
- 1.5 Preparazione del documento di specifiche
  - 1.5.1 Definizione del ciclo di sviluppo da adottare
  - 1.5.2 Definizione delle fasi del progetto
- 1.6 Definizione e struttura del team
  - 1.6.1 Definizione di tipologia di team
  - 1.6.2 Definizione dei metodi di comunicazione
  - 1.6.3 Definizione delle responsabilità e persone di riferimento
- 1.7 Stima dei costi
  - 1.7.1 Definizione dei costi diretti
  - 1.7.2 Definizione dei costi indiretti
- 1.8 Gestione dei rischi
  - 1.8.1 Identificazione dei rischi di progetto
  - 1.8.2 Definizione delle probabilità e dell'impatto
  - 1.8.3 Definizione del piano di gestione dei rischi
- 1.9 Pianificazione del lavoro
  - 1.9.1 Definizione dell'attività e creazione WBS
  - 1.9.2 Assegnazione delle risorse ad ogni attività
  - 1.9.3 Definizione delle Milestones
  - 1.9.4 Stime dei tempi
  - 1.9.5 Previsione delle date di consegna dei documenti di specifica
  - 1.9.6 Proiezione dei costi
  - 1.9.7 Definizione del budget
- 1.10 Stesura e revisione documentale

##### 2. Progettazione concettuale

- 2.1 Analisi dei requisiti
  - 2.1.1 Analisi dei requisiti funzionali
    - 2.1.1.1 Analizzare organizzazione utente



- 2.1.1.2 Analizzare interfacce utente
    - 2.1.1.3 Analizzare ambiente di accoglienza
    - 2.1.1.4 Analizzare modalità di navigazione
  - 2.1.2 Analisi dei requisiti non funzionali
    - 2.1.2.1 Analizzare prestazioni richieste
    - 2.1.2.2 Analizzare livello di servizio atteso
    - 2.1.2.3 Analizzare usabilità
  - 2.2 Analisi dei dati
    - 2.2.1 Identificazione di tutte le entità
  - 2.3 Analisi del design
    - 2.3.1 Analisi dell'interfaccia grafica
      - 2.3.1.1 Studio delle linee guida
      - 2.3.1.2 Analisi Concorrenziale
      - 2.3.1.3 Progettazione layout
      - 2.3.1.4 Progettazione prototipo
- 3. Progettazione tecnica**
  - 3.1 Verifica delle specifiche funzionali
  - 3.2 Scelta dell'ambiente di sviluppo
    - 3.2.1 Scelta del software di supporto
    - 3.2.2 Scelta dei software di versioning
    - 3.2.3 Scelta aggiuntive
  - 3.3 Organizzazione del codice
    - 3.3.1 Divisione in moduli
- 4. Sviluppo**
  - 4.1 Verifica progettazione tecnica
  - 4.2 Sviluppo Codice
  - 4.3 Documentazione
- 5. Test**
  - 5.1 Test di modulo
    - 5.1.1 Preparazione casi di prova
    - 5.1.2 Tracciare esito test
  - 5.2 Test di integrazione
    - 5.2.1 Preparare piano di test
    - 5.2.2 Preparare casi di prova
    - 5.2.3 Preparare ambiente di prova
    - 5.2.4 Tracciare esito test
  - 5.3 Test di stress
    - 5.3.1 Tracciare esito test
  - 5.4 Pubblicazione versione beta
- 6. Rilascio**
  - 6.1 Rilascio su Play Store
    - 6.1.1 Ottenere account Play Store

- 6.1.2 Preparazione pagina dell'App
- 6.1.3 Rilascio dell'App
- 6.2 Gestione della configurazione
  - 6.2.1 Definizione della politica della manutenzione
  - 6.2.2 Definizione delle regole di gestione delle versioni
- 6.3 Creazione Manuali

## 7. Pianificazione e controllo progetto

- 7.1 Controllo della qualità
- 7.2 Controllo dei costi
- 7.3 Controllo nell'ambito del progetto
- 7.4 Riunioni di coordinamento con il team
- 7.5 Verifica rispetto le tempistiche
- 7.6 Verifiche con il committente

### 5.1.2 Attività e funzioni

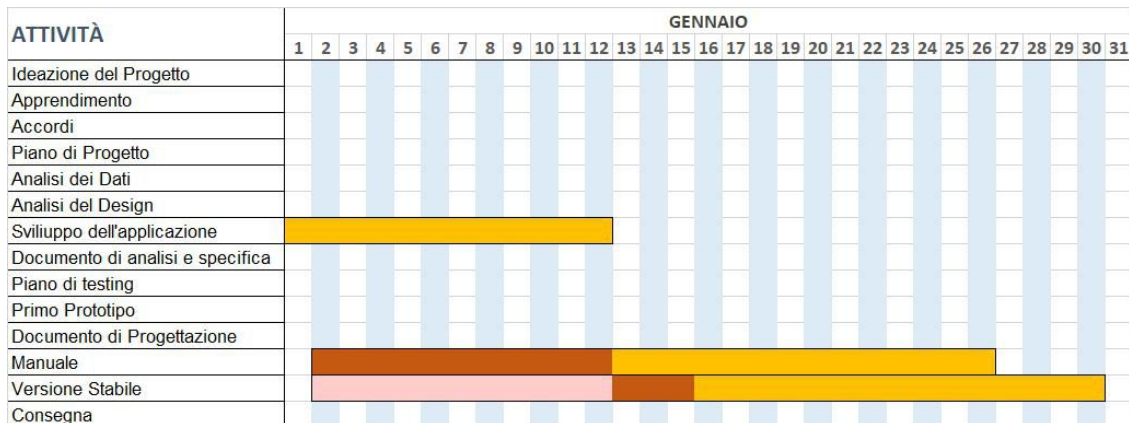
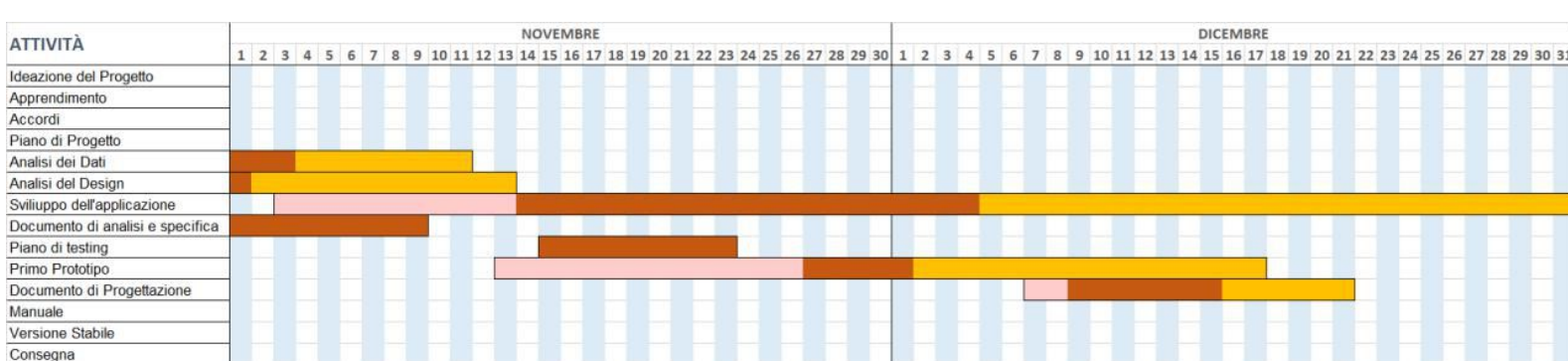
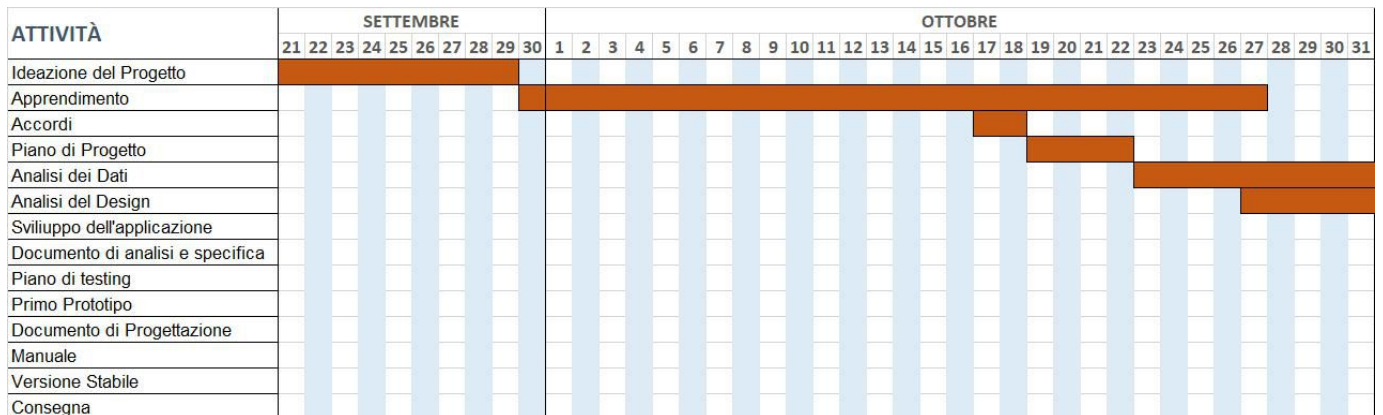
ATTIVITÀ	INIZIO	FINE	DURATA PREVISTA GG	DURATA EFFETTIVA GG	DIFFERENZA GG	ORE/UOMO PREVISTE	ORE/UOMO EFFETTIVE	DIFFERENZA ORE/UOMO
Ideazione del Progetto	21/09/2017	29/09/2017	8	8	0	30	24	-6
Apprendimento	30/09/2017	27/10/2017	27	27	0	70	79	9
Accordi	17/10/2017	18/10/2017	1	1	0	4	4	0
Piano di Progetto	18/10/2017	22/10/2017	4	4	0	20	12	-8
Analisi dei Dati	23/10/2017	11/11/2017	11	19	8	20	25	5
Analisi del Design	27/10/2017	13/11/2017	5	17	12	10	20	10
Sviluppo dell'applicazione	14/11/2017	12/01/2018	31	59	28	60	75	15
Documento di analisi e specifica	01/11/2017	09/11/2017	8	8	0	32	26	-6
Piano di testing	15/11/2017	23/11/2017	8	8	0	32	20	-12
Primo Prototipo	27/11/2017	17/12/2017	18	20	2	20	15	-5
Documento di Progettazione	09/12/2017	21/12/2017	8	12	4	25	29	4
Manuale	02/01/2018	26/01/2018	10	24	14	15	10	-5
Versione Stabile	13/01/2018	30/01/2018	13	17	4	26	38	12
Consegna	30/01/2018	30/01/2018	0	0	0	0	0	0

La tabella mostra la durata di ogni attività con obiettivo ultimo la buona riuscita del progetto entro la data di Consegna.




Ogni attività ha una data di inizio e una data di fine, la durata in giorni, in ore/uomo e la differenza rispetto al piano iniziale: sono evidenziate in rosso le caselle che hanno avuto un aumento tra il dato previsto e il dato effettivo; al contrario, sono evidenziate in verde le caselle in cui c'è stata una realizzazione più veloce del previsto.

Si può osservare che non a tutti i giorni corrispondono le stesse ore/uomo, questo perché non tutte le risorse umane sono state utilizzate per 8 ore al giorno. Infatti, la durata prevista e la durata effettiva sono espresse in GIORNI (GG) e non in GIORNI/UOMO, che corrisponderebbero a 8 ore/uomo per giorno.

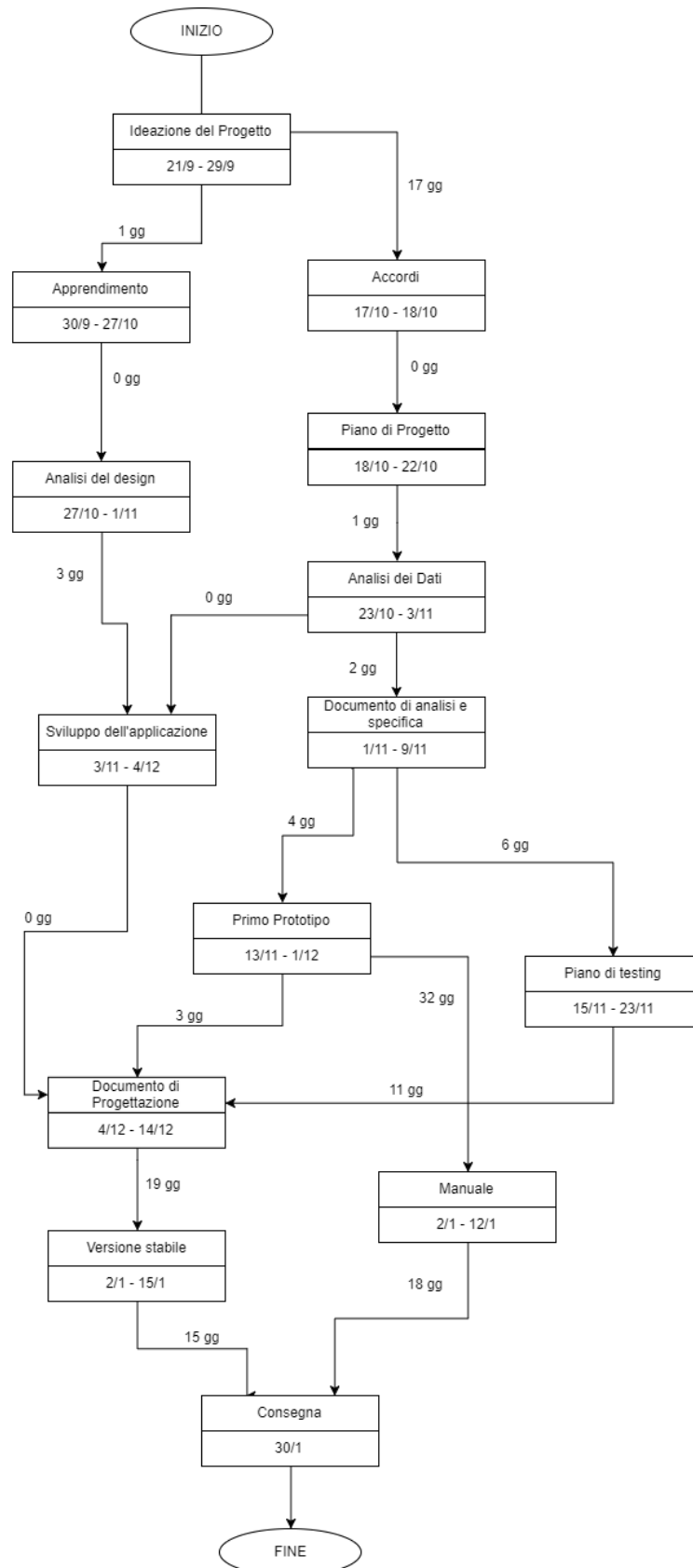
### 5.1.3 Gantt



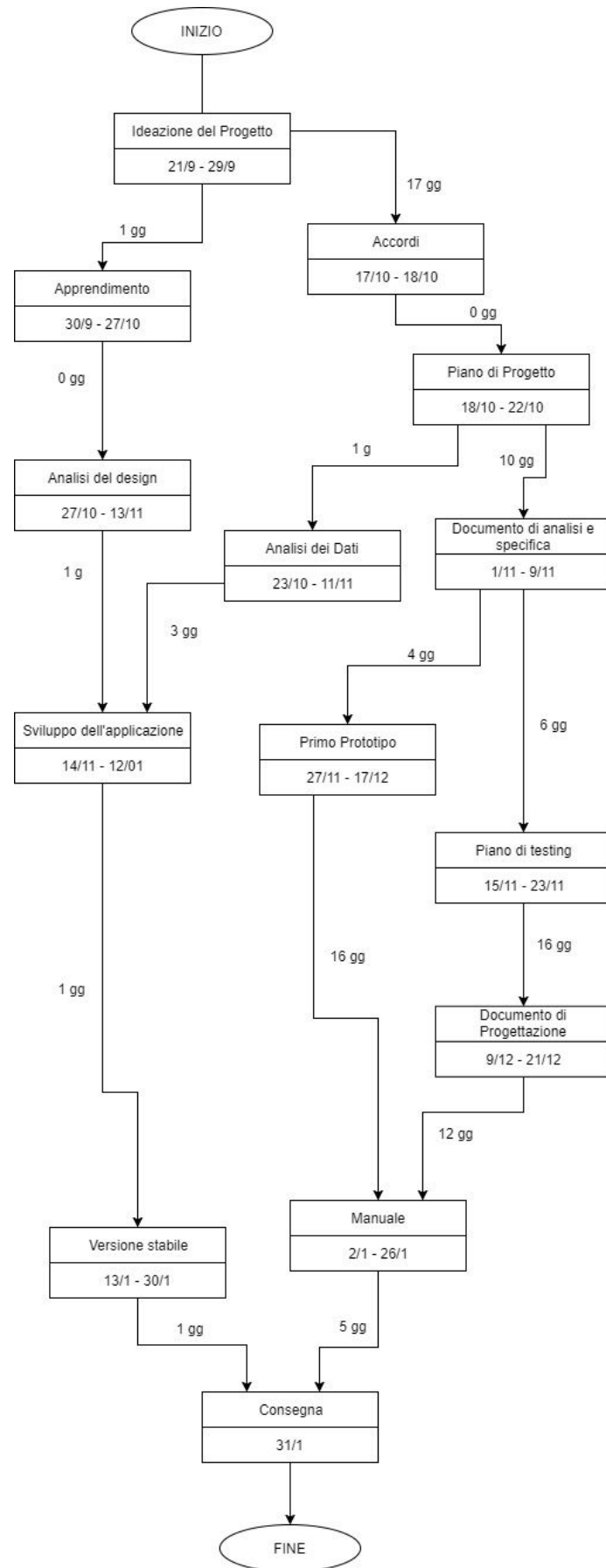
LEGENDA:

-  periodo attività stabilita
-  periodo attività di ritardo rispetto a quanto prestabilito
-  periodi attività in aggiunta

## 5.1.4 Pert



Pert v1



Pert v2

## **5.2 Dipendenze**

Le dipendenze principali sono date dalle macro-task, ossia i documenti che accompagnano le varie fasi di progetto; ogni documento sarà diviso in sottosezioni.

Utilizzando il modello evolutivo, ogni attività sarà strettamente vincolata alla precedente, in quanto i task e le possibili migliorie all'applicazione dovranno essere discussi fase per fase sempre considerando i dati in nostro possesso.

## **5.3 Risorse Necessarie**

I membri del team sono stati scelti anche in base alle competenze condivise e non; ciò permette una suddivisione del lavoro equa ma anche ottimizzata. Infatti, tutti sono in grado di svolgere la maggior parte dei task, ma le competenze specifiche di ogni membro in un certo ambito permettono una maggiore qualità della relativa fase o sottofase di progetto. Ne consegue che tutti i membri del gruppo risultano fondamentali per il successo, ma la loro poliedricità permette anche un ottimo grado di tolleranza.

## **5.4 Allocazione del Budget e delle Risorse**

Non sono previsti budget per lo sviluppo dell'applicazione in quanto i software utilizzati sono gratuiti, fatta eccezione per le ore/uomo per portare a termine il progetto, che sono a carico del team. La frequenza delle ore di sviluppo varierà in base agli impegni universitari e non dei componenti.

## **5.5 Pianificazione**

Il progetto ha una serie di scadenze definite dal committente, ossia:

- 1- Piano di Progetto (23/10/2017);
- 2- Documento di Analisi e Specifica (09/11/2017);
- 3- Piano di Testing (23/11/2017);
- 4- Documento di Progettazione (15/12/2017);
- 5- Realizzazione e messa in linea (31/01/2018).

Per la pianificazione di task più precisi sono necessari i dati che verranno dall'ufficio open data di Ca' Foscari; una volta visionati ed analizzati sarà possibile studiare un piano più specifico e, conseguentemente, stabilire le milestones che dovranno accompagnare lo sviluppo del progetto.