

PIANO DI PROGETTO

GRUPPO: ArceTeam

COMPONENTI GRUPPO:

#876589, CASAROTTI GIULIO
#875921, FERRARI SIMONE
#874214, GALLO GIULIA
#875926, TROLESE GIULIO



**21 Dicembre 2020,
Versione 2.0**

**Corso di Ingegneria del Software 2020
UNIVERSITÀ CA' FOSCARI DI VENEZIA**

SOMMARIO

[1] INTRODUZIONE	3
[1.1] Overview del Progetto	3
[1.2] Deliverables del Progetto	3
[1.3] Evoluzione del Progetto	4
[1.4] Materiale di riferimento	4
[1.5] Definizioni e Abbreviazioni	4
[2] ORGANIZZAZIONE DEL PROGETTO	7
[2.1] Modello del Processo	7
[2.2] Struttura Organizzativa	7
[2.3] Interfacce Organizzative	7
[2.4] Responsabilità di Progetto	7
[3] PROCESSI GESTIONALI	8
[3.1] Obiettivi e Priorità	8
[3.2] Assunzioni, Dipendenze, Vincoli	8
[3.3] Gestione dei rischi	8
[3.4] Meccanismi di monitoraggio e di controllo	10
[3.5] Pianificazione dello staff	10
[4] PROCESSI TECNICI	11
[4.1] Metodi, Strumenti e Tecniche	11
[4.2] Documentazione del Software	11
[4.3] Funzionalità di supporto al progetto	11
[5] PIANIFICAZIONE DEL LAVORO, DELLE RISORSE UMANE E DEL BUDGET	12
[5.1] WBS (Work Breakdown Structure)	12
[5.2] Dipendenze	13
[5.2.1] Diagramma di Pert (Figura 3)	14
[5.2.2] Diagramma di Gantt (Figura 4)	15
[5.3] Risorse Necessarie	16
[5.4] Allocazione del Budget e delle Risorse	16
[5.5] Pianificazione	16
NOTA RIGUARDANTE LA FASE DI TESTING	18

PIANO DI PROGETTO		
DATA	VERSIONE	DESCRIZIONE
18/10/2020	1.0	Prima stesura del documento
21/12/2020	2.0	Modifica date deliverables, completato parti mancanti, nota di testing

[1] INTRODUZIONE

[1.1] Overview del Progetto

Il progetto proposto consiste in un'applicazione per dispositivi mobili Android che permetta di **mantenere uno storico di 7 giorni di tutte le persone che si sono incrociate lungo la strada e nei luoghi frequentati.**

In particolare, la nostra applicazione dovrebbe essere in grado di **scambiare informazioni** con i dispositivi che entrano nel raggio d'azione del **Bluetooth Low Energy** del nostro smartphone.

Che informazioni deve condividere?

La risposta è: **cioè che si vuole.**

In ogni caso, identifichiamo **due grandi gruppi di informazioni:**

- **Obbligatorie:** Data, Ora e Luogo dello scambio, Nickname, Sesso, Anno di nascita, Foto profilo;
- **Facoltative:** Nome, Cognome, Data di nascita completa, Città di residenza, Contatti vari (email, numero di telefono, profilo Instagram, Linkedin, ...), segni particolari (occhiali, colore capelli, ...), e qualsiasi altra informazione si voglia condividere.

Ovviamente, per questioni di **privacy**, non obblighiamo nessun utente a **condividere informazioni** strettamente **personalì**: sarà lui stesso a decidere cosa rendere pubblico, e cosa no.

Piccola nota sulla **foto profilo**: essa risulta essere **l'elemento chiave** per il riconoscimento di una persona: stiamo facendo una "ricerca inversa", ovvero, non sono a conoscenza della sua identità, ma la so riconoscere dal volto.

In altre parole, la classica ricerca che avviene nei social network, è della forma:
"**Identità→Foto**"; noi, invece, proponiamo l'inverso: "**Foto→Identità**".

Possibili applicazioni:

- **Studente in campus:** potrebbe essere utile poter contattare, a posteriori, una persona presente in aula alla stessa lezione per chiedere appunti o altre informazioni.
- **Svago:** capita di sedersi al tavolo di un bar e notare una persona interessante, ma per motivi *x* non vi è la possibilità di interagirci al momento: grazie alla nostra app, sarà possibile avere i suoi contatti e mettersi in comunicazione!
- **Ambito lavorativo:** dopo un appuntamento con un cliente, ci accorgiamo di aver perso il post-it contenente le annotazioni di numero di telefono e email: possiamo recuperarle dalla nostra app!

[1.2] Deliverables del Progetto

Le **scadenze dei deliverable** del progetto sono attualmente fissate nelle seguenti date:

- Definizione **gruppi di lavoro** entro il 1 Ottobre 2020
- Proposta di progetto entro il 1 Ottobre 2020
- Piano di progetto entro il 18 Ottobre 2020
- Documento di **analisi e specifica** entro il 02 Novembre 2020
- Piano di testing entro il 19 Novembre 2020
- Documento di **progettazione** entro il 14 Dicembre 2020
- Codice e messa in linea del sistema entro il 15 Gennaio 2021

Nota: le scadenze, durante il corso, potrebbero subire lievi variazioni.

[1.3] Evoluzione del Progetto

Abbiamo **proposto** nella [1.1] **tre esempi applicativi**, ma ovviamente le occasioni d'uso sono pressoché infinite.

Ad esempio, al giorno d'oggi, la nostra applicazione potrebbe integrare funzionalità alla pari di **Immuni**, l'app per il tracciamento dei contagi di **Covid-19**.

In futuro, i dati raccolti e condivisi dall'applicazione potrebbero essere salvati in un database per poi utilizzarli per guidare **analisi statistiche**, come ad esempio:

- Conoscere gli orari in cui un determinato luogo è più o meno affollato;
- Valutare se un determinato luogo è frequentato maggiormente da giovani o anziani;
- Regolare i flussi turistici basandosi sul luogo di provenienza e consigliando le attrazioni di maggior interesse;
- Proporre suggerimenti di locali a noi sconosciuti ma compatibili con le nostre abitudini.

[1.4] Materiale di riferimento

Per sviluppare la nostra applicazione faremo riferimento alla [documentazione ufficiale Android](#).

Inoltre, per risolvere eventuali dubbi, ci appoggiamo al [corso ufficiale di google](#) per la programmazione Android in Java.

Potranno tornare utili, inoltre, le famose community di [StackOverflow](#) e di [GitHub](#).

[1.5] Definizioni e Abbreviazioni

Android: sistema operativo mobile basato su kernel linux e sviluppato da Google.

Android Studio: IDE di riferimento per lo sviluppo del software Android.

API (Application Programming Interface): set di definizioni e protocolli con i quali vengono realizzati e integrati software applicativi.

Best-practices: metodo o una tecnica che è stata generalmente accettata come superiore a qualsiasi alternativa perché produce risultati che sono superiori a quelli raggiunti con altri mezzi o perché è diventata un modo standard di fare le cose.

Bluetooth Low Energy (BLE): a differenza del Bluetooth classico, il Bluetooth Low Energy (BLE) è progettato per fornire un consumo energetico notevolmente inferiore.

Campus: il complesso degli edifici e dei terreni di un'università.

Covid-19: la malattia provocata dal nuovo Coronavirus prende il nome: "COVID-19" (dove "CO" sta per corona, "V" per virus, "D" per disease e "19" indica l'anno in cui si è manifestata).

Deadlines: scadenze.

Deliverables: oggetto materiale o immateriale realizzato (fornito/consegnato) come risultato di un'attività del progetto.

Democratico Decentralizzato: è una struttura organizzativa nella quale vi è un'assenza di un leader specifico privilegiando la comunicazione paritetica all'interno del gruppo.

Diagrams.net: spazio di lavoro dedicato alla realizzazione di schemi e grafici.

Diagramma di Gantt: diagramma a barre che relaziona le attività con il tempo, visualizzando la durata di ogni attività e la sua allocazione nel tempo.

Diagramma di Pert (Program Evaluation Review Technique): è un grafo nel quale i nodi rappresentano le scadenze o i punti cardine e le frecce rappresentano le dipendenze tra le attività.

Discord: software VOIP (Voice Over IP) gratuito disponibile su desktop e mobile.

Features: caratteristiche uniche di un prodotto o servizio.

Feedback: elenco contenente possibili migliorie e segnalazioni di errori, inviato allo sviluppatore di un'applicazione da un utente/tester che la collauda.

Foto→Identità: a partire dall'aspetto fisico di una persona (Foto) si risale ai dati personali (Nome, Cognome, ...).

GitHub: servizio di hosting per progetti software. Il nome deriva dal fatto che "GitHub" è una implementazione dello strumento di controllo versione distribuito Git.

Google Drive: luogo di salvataggio e backup condiviso basato su Cloud (accessibile dalla rete);

Google Docs: editor di testo di Google utile alla stesura di documenti.

Google Sheets: foglio di calcolo di Google.

GUI (Graphical User Interface): tipologia di interfaccia utente che consente l'interazione uomo-macchina in modo visuale utilizzando rappresentazioni grafiche.

Hardware: è la parte materiale di un computer, ovvero tutte quelle parti elettroniche, elettriche, meccaniche, magnetiche, ottiche che ne consentono il funzionamento.

IDE (Integrated Development Environment): ambiente di sviluppo che, in fase di programmazione, supporta i programmati nello sviluppo e debugging del codice sorgente di un programma.

Identità→Foto: a partire da un insieme di dati conosciuti riguardo una persona (Nome, Cognome, ...) si risale al suo aspetto fisico.

Immuni: applicazione ufficiale dello Stato Italiano per il tracciamento dei contagi Covid-19.

Interfaccia: dispositivo di collegamento in grado di assicurare la comunicazione tra due sistemi informatici altrimenti incompatibili.

Java: linguaggio di programmazione orientato ad oggetti.

Marshmallow 6.0: è il nome in codice per la release 6.0 del sistema operativo Android.

Milestones: è un evento significativo, all'interno del progetto, che sta ad indicare il completamento di un "deliverable" principale e/o un passaggio di "stato" del progetto.

Modello a cascata: modello di processo nel quale si stabiliscono a priori le varie fasi di sviluppo.

Nickname: nome abbreviato o soprannome assegnato a un utente.

Overview: breve introduzione che riassume il contenuto del documento.

P2P (Peer to Peer): nelle telecomunicazioni indica un modello di architettura logica di rete informatica in cui i nodi non sono gerarchizzati unicamente sotto forma di client o server fissi ('clienti' e 'serventi'), ma anche sotto forma di nodi equivalenti o 'paritari' (peer), potendo fungere al contempo da client e server verso gli altri nodi terminali (host) della rete.

Project Management: l'applicazione di conoscenze, capacità, strumenti e tecniche alle attività di Progetto per soddisfarne i requisiti.

Software: l'insieme delle componenti immateriali (strato logico/intangibile) di un sistema elettronico di elaborazione; è contrapposto all'hardware.

Task: compito specifico di un programma applicativo.

Telegram: software di messaggistica istantanea gratuito disponibile per i principali sistemi operativi.

Tester: persone aventi ruolo di verificare il corretto funzionamento del prodotto.

Trello: software di Project Management per la gestione del lavoro di gruppo.

UML (Unified Modeling Language): linguaggio di modellazione e di specifica basato sul paradigma orientato agli oggetti.

VOIP (Voice Over IP): indica una tecnologia che rende possibile effettuare una conversazione, analoga a quella che si potrebbe ottenere con una rete telefonica, sfruttando una connessione Internet.

WBS (Work Breakdown Structure): è una scomposizione gerarchica orientata ai deliverable del lavoro che deve essere eseguito dal gruppo di progetto per conseguire gli obiettivi prestabiliti.

[2] ORGANIZZAZIONE DEL PROGETTO

[2.1] Modello del Processo

Per sviluppare il nostro progetto, seguiremo il “**Modello a cascata**”.

Tale modello si basa sul principio di **stabilire a priori le varie fasi di sviluppo** del sistema: questo permette di avere un **progetto ben strutturato** e **minimizza i cambiamenti in corso d'opera**. Abbiamo scelto questa organizzazione in quanto **la nostra filosofia** è:

“Femo ‘na roba ‘na volta, ma ea femo ben!”¹

Partiremo da un’**analisi e definizione dei requisiti** per arrivare alla fase finale di **installazione e mantenimento del sistema**. Nella fase intermedia, **progetteremo il software**, **implementeremo e testeremo le singole unità** che compongono il sistema e, per concludere, **integreremo ogni parte e testeremo tutto l’insieme**.

[2.2] Struttura Organizzativa

La struttura organizzativa adottata è di tipo “**Democratico Decentralizzato**”.

Il motivo di tale scelta è dovuto al numero ridotto di persone che compongono il gruppo.

Questo modello **non evidenzia un leader specifico**, piuttosto tutti i **componenti collaborano tra loro** e comunicano in modo orizzontale.

Le varie fasi di sviluppo sono comunque assegnate a dei **sottogruppi** in base alle skills di ognuno, mantenendo una salda comunicazione tra tutti i membri.

La comunicazione avviene principalmente mediante un gruppo **Telegram** e un server **Discord**. Inoltre, abbiamo a disposizione uno spazio condiviso su **Google Drive** per lavorare simultaneamente sui documenti e utilizziamo una repository **GitHub** per il versioning del codice sorgente.

Per il **Project Management**, utilizziamo un software dedicato chiamato **Trello**, il quale ci permette di organizzare il gruppo e tenere traccia dei progressi.

[2.3] Interfacce Organizzative

Durante lo sviluppo, ci relazioneremo con:

- Il Prof. Agostino Cortesi e il Tutor **Giulio Zausa** per informazioni sul progetto;
- Il Dott. Alvise Spanò per dubbi e problemi relativi al codice Android;
- Eventuali **Tester** per ottenere feedback sull’usabilità dell’applicazione.

[2.4] Responsabilità di Progetto

A livello organizzativo, il **capogruppo** è **Ferrari Simone** il quale ha il ruolo di **portavoce**. Dato che adottiamo una struttura organizzativa di tipo “**Democratico Decentralizzato**”, il **capogruppo** è solo **simbolico** e non rappresenta un leader vero e proprio, pertanto la responsabilità ricade su tutti i componenti.

In ogni caso, ognuno di noi ha diverse **capacità distintive**, per cui cercheremo di sfruttare suddividendoci il lavoro nella maniera più adeguata.

¹ **Traduzione:** Facciamo una cosa una volta sola, ma la facciamo bene.

[3] PROCESSI GESTIONALI

[3.1] Obiettivi e Priorità

L'obiettivo del progetto consiste nello sviluppo di un'applicazione Android P2P in linea con le specifiche date e la redazione di tutti i documenti di accompagnamento necessari.

Le nostre **priorità** sono:

- Portare a termine il progetto entro i termini specificati in [1.2];
- Rendere più semplice la comunicazione con persone a noi sconosciute;
- Mettere sul mercato un software funzionante e innovativo;
- Mantenere il gruppo unito, garantendo un'alta collaborazione;
- Ottenere un livello di conoscenza adeguato della programmazione Android per lo sviluppo del software.

[3.2] Assunzioni, Dipendenze, Vincoli

Per il problema della **privacy**, assumiamo che un utente della nostra applicazione sia consapevole che i dati condivisi siano esclusivamente quelli che lui stesso ha deciso di rendere disponibili.

Inoltre, per garantire l'**utilità dell'applicazione**, vi è una forte necessità di un alto numero di download, in quanto i dati disponibili per la condivisione aumentano in funzione del numero di utilizzatori.

Per poter **usufruire del nostro prodotto**, sarà necessario avere a disposizione un dispositivo mobile Android, con una versione almeno pari a **Marshmallow 6.0** (aumentata da 5.0 per adeguamenti relativi al Bluetooth).

Un aspetto fondamentale per la **riuscita del progetto** è la conoscenza della programmazione Android da parte dei membri del gruppo.

Un vincolo fondamentale è il **tempo a disposizione** per il completamento del progetto.

[3.3] Gestione dei rischi

Per garantire una corretta riuscita del progetto, è fondamentale **individuare** a priori il maggior numero di **rischi** possibili per poter pianificare **azioni risolutive** adeguate.

I **rischi individuati** sono i seguenti:

N°	Rischio	Probabilità	Impatto	Azione
1	Perdita del codice	Bassa	Alto	PREVENZIONE: Manteniamo più copie in più dispositivi di tutto il lavoro eseguito. Inoltre, lo rendiamo disponibile anche in cloud (Google Drive, GitHub) in modo tale da poterlo recuperare.
2	Possibile lockdown che ci impedisca di testare l'app	Media	Medio-Alto	MITIGAZIONE: Testeremo l'app all'interno dei rispettivi nuclei familiari.

3	Salute dei membri	Medio-Bassa	Medio	MITIGAZIONE: Il lavoro della persona malata verrà portato avanti dagli altri membri del gruppo.
4	Programmi di comunicazione non funzionanti	Bassa	Alto	PREVENZIONE: Utilizziamo diversi software per comunicare. SOLUZIONE: Utilizzare altri software.
5	Problemi di comunicazione con i relatori	Bassa	Medio	SOLUZIONE: Contattare più volte e in modo diverso i relatori.
6	Mancanza di un numero adeguato di tester	Media	Medio-Alto	PREVENZIONE: Iniziare fin da subito la ricerca di futuri tester. SOLUZIONE: Attrezzarci di più dispositivi Android per testare in autonomia.
7	Rottura di componenti hardware	Bassa	Medio-Basso	SOLUZIONE: Utilizzare altri mezzi.
8	Difficoltà realizzative di alcune features	Medio-Bassa	Alto	PREVENZIONE: Assicurarsi già in fase di pianificazione la fattibilità delle Features
9	Difficoltà con la programmazione Android	Media	Medio-Alto	MITIGAZIONE: Cercare risposte su internet, dai manuali o dai professori.
10	Non rientrare nelle tempistiche richieste	Medio-Bassa	Medio-Alto	PIANIFICAZIONE: Pianifichiamo fin da subito le varie tempistiche cercando di rispettarle. SOLUZIONE: Riprogrammare il lavoro.
11	Non individuare bug "leggeri"	Alta	Basso	PREVENZIONE: testare di volta in volta le nuove features inserite SOLUZIONE: Correggere il bug e aggiornare il software.
12	Non individuare bug "pesanti"	Medio-Bassa	Alto	PREVENZIONE: testare di volta in volta le nuove features inserite SOLUZIONE: Correggere il bug e aggiornare il software.
13	Conflitti interni al team	Bassa	Medio	PREVENZIONE: Ascoltarsi reciprocamente e trovare compromessi.
14	Rottura del gruppo	Bassa	Alto	PREVENZIONE: Fare il possibile per restare uniti.
15	Problemi di varia natura che impediscono la presenza di un componente del gruppo	Medio-Alta	Medio-Basso	MITIGAZIONE: Il lavoro della persona assente verrà portato avanti dagli altri membri del gruppo.

Figura 1: Tabella dei rischi

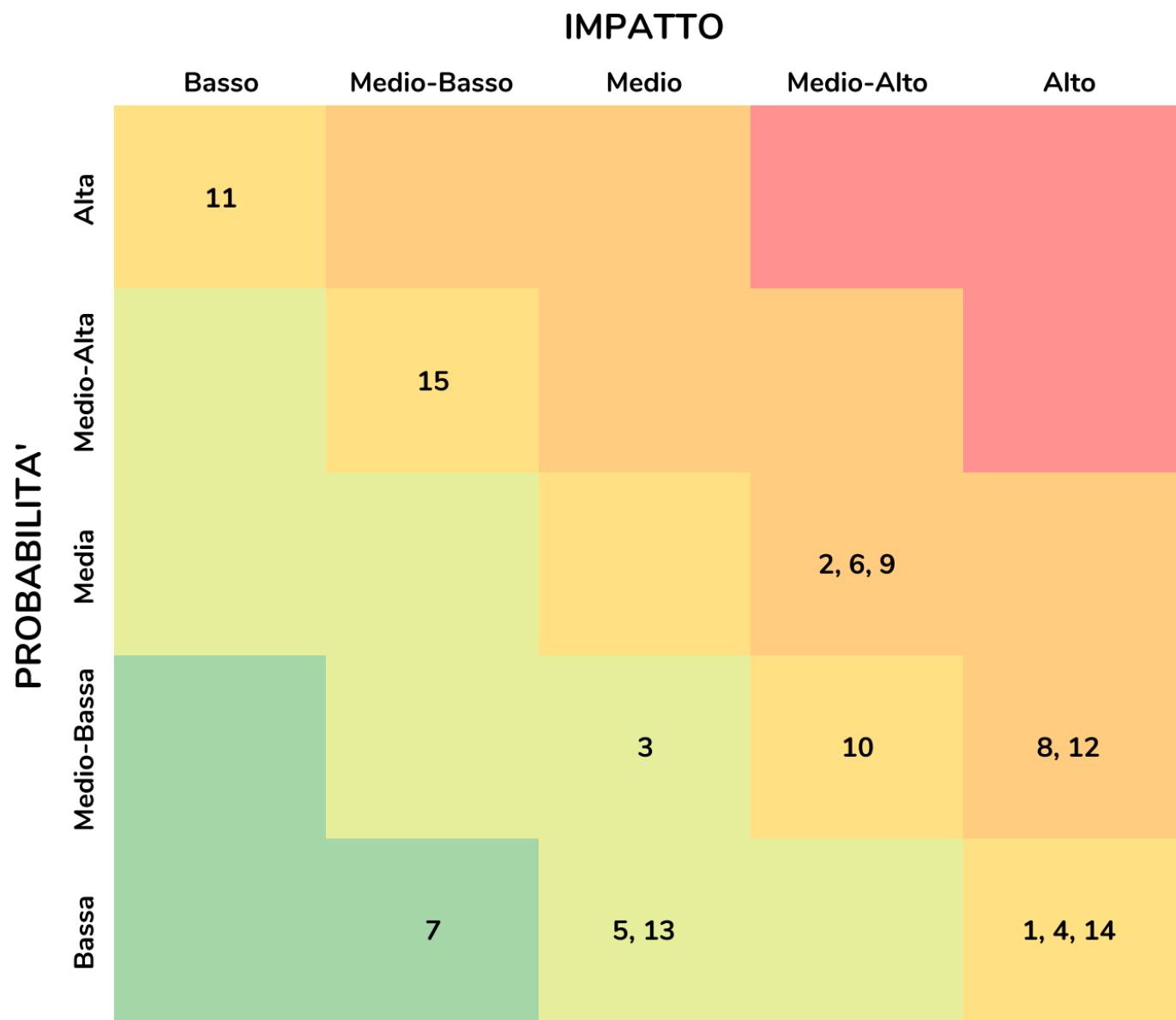


Figura 2: Matrice Probabilità-Impatto dei rischi

[3.4] Meccanismi di monitoraggio e di controllo

Il gruppo resterà **costantemente in contatto** mediante i software citati al [2.2].

Inoltre, a scopo di **Mantenere il gruppo unito**, organizzeremo attività terze di puro svago.

Periodicamente verranno **programmate delle date** il cui tema sarà la **revisione del lavoro** svolto fino a tale data, valutando possibili cambiamenti.

I **progressi ottenuti** saranno continuamente tenuti **sotto controllo** mediante software di versioning, quale GitHub e un software di Project Management chiamato Trello.

[3.5] Pianificazione dello staff

Il lavoro verrà principalmente eseguito **collaborando** tutti simultaneamente.

Nonostante ciò, alcune attività potrebbero essere più indicate al **singolo membro** o ad un **sottogruppo**, in modo da sfruttare al meglio il tempo e le risorse umane.

[4] PROCESSI TECNICI

[4.1] Metodi, Strumenti e Tecniche

La **collaborazione di gruppo** è alla base del nostro progetto, anche se non si escludono mansioni assegnate al **singolo** od a **sottogruppi**.

Il **lavoro** avverrà principalmente da **remoto**, data la situazione attuale, ma, se possibile, organizzeremo delle sessioni in presenza per effettuare testing.

Faremo uso di diversi **strumenti**, quali:

- **Android Studio**: IDE per lo sviluppo del software;
- **Google Drive**: luogo di salvataggio e backup condiviso; offre, inoltre:
 - **Google Docs**: editor di testo per la stesura di documenti;
 - **Google Sheets**: foglio di calcolo per la realizzazione di tabelle;
- **Diagrams.net**: spazio di lavoro dedicato alla realizzazione di schemi e grafici;
- **GitHub**: servizio utile ad eseguire il versioning e backup del codice;
- **Trello**: software di Project Management per la gestione del gruppo;
- **PC**: ognuno di noi lavorerà sul proprio computer sui cui vi è installato Windows/MacOS;
- **Dispositivi mobili Android**: per il test dell'applicazione utilizzeremo più dispositivi mobili android.

[4.2] Documentazione del Software

La **documentazione** che accompagnerà il nostro progetto e le relative **date di consegna** sono le seguenti:

- **Piano di progetto** entro il 18 Ottobre 2020
- **Documento di analisi e specifica** entro il 02 Novembre 2020
- **Piano di testing** entro il 19 Novembre 2020
- **Documento di progettazione** entro il 14 Dicembre 2020

Durante **l'evolversi del progetto**, i documenti verranno **aggiornati** rendendo i dettagli realizzativi più precisi.

[4.3] Funzionalità di supporto al progetto

Per garantire una **buona qualità** del progetto, dobbiamo rispettare i seguenti punti:

- **Aggiornare** costantemente la **documentazione**;
- **Rispettare** le **tempistiche** che specificheremo nei punti a venire;
- **Strutturare** il **codice** in modo chiaro e ben organizzato già a partire dalle prime fasi;
- **Rispettare** le **best-practices** della programmazione Android;
- **Mantenere** un livello di **collaborazione** elevato all'interno del gruppo;
- Attenersi al **modello di processo stabilito** in partenza e svolgere le tasks nell'ordine prestabilito;
- **Testare** sistematicamente l'applicazione in tutte le sue parti;
- Raccogliere **feedback** da persone terze. **ATTENZIONE**: [vedi nota a fine documento](#).

[5] PIANIFICAZIONE DEL LAVORO, DELLE RISORSE UMANE E DEL BUDGET

[5.1] WBS (Work Breakdown Structure)

1. Fase preliminare

1. Definizione idea iniziale di progetto;
 1. Individuazione del tema del progetto
 2. Verifica dell'originalità dell'idea
 3. Individuazione delle risorse e delle competenze necessarie
 4. Analisi di fattibilità
2. Definizione degli obiettivi del progetto;
 1. Individuazione di una potenziale clientela
3. Presentazione idea al committente;

2. Documentazione

1. Piano di progetto
 1. Descrizione di massima del progetto
 1. Breve introduzione
 2. Stesura delle funzionalità dell'applicazione
 3. Ricerca del materiale a cui fare riferimento
 4. Proposte per lo sviluppo futuro
 2. Definizione della struttura organizzativa
 3. Definizione del modello di processo
 4. Analisi e gestione dei rischi
 1. Identificazione dei rischi
 2. Assegnamento della probabilità e dell'impatto
 3. Suggerimento tecniche di gestione dei rischi
 5. Processi tecnici
 6. Costruzione del WBS
 7. Diagramma di Pert
 8. Diagramma di Gantt
 9. Analisi e stima dei costi
 10. Revisione e consegna
2. Analisi dei requisiti
 1. Analisi dei requisiti funzionali
 2. Analisi dei requisiti non funzionali
 3. Analisi dei requisiti di implementazione, prestazione e usabilità
 1. Studio linee guida Android
 2. Studio delle API Android adeguate allo scopo
 4. Definizione modelli del sistema
 1. Definizione delle componenti in gioco
 2. Diagramma UML
 5. Verifica dei requisiti e della loro fattibilità
 6. Prototipazione della GUI
3. Piano di Testing
 1. Definizione delle modalità di testing
 2. Definizione degli elementi da testare
4. Documento di progettazione

3. Progettazione tecnica

1. Scelta degli strumenti di Project Management
2. Definizione ambiente di sviluppo
3. Scelta delle API Android da utilizzare per lo sviluppo
4. Scelta delle API Terze
5. Organizzazione della struttura fisica del progetto (Directory, codice, ...)

4. Sviluppo

1. Sviluppo GUI
2. Sviluppo delle features
3. Documentazione

5. Testing finale

1. Ricerca di Tester
2. Individuazione casi di prova
3. Testing e raccolta feedback
4. Monitoraggio dell'esito dei test
5. Pianificazione e applicazione di eventuali fix

6. Rilascio

1. Revisione della documentazione
2. Rilascio dell'applicazione

7. Discussione del progetto

1. Consegna del progetto
2. Discussione del progetto

[5.2] Dipendenze

Qui nel seguito, analizziamo le dipendenze proponendo un **Diagramma di Pert**.

Successivamente, mediante un **Diagramma di Gantt** visualizziamo l'organizzazione temporale del progetto.

Dato che gli schemi sottostanti risultano essere abbastanza dettagliati, può risultare difficoltosa la loro visualizzazione su questo documento, pertanto **alleghiamo**, ai **seguenti link**, i file in questione.

Diagramma di Pert:

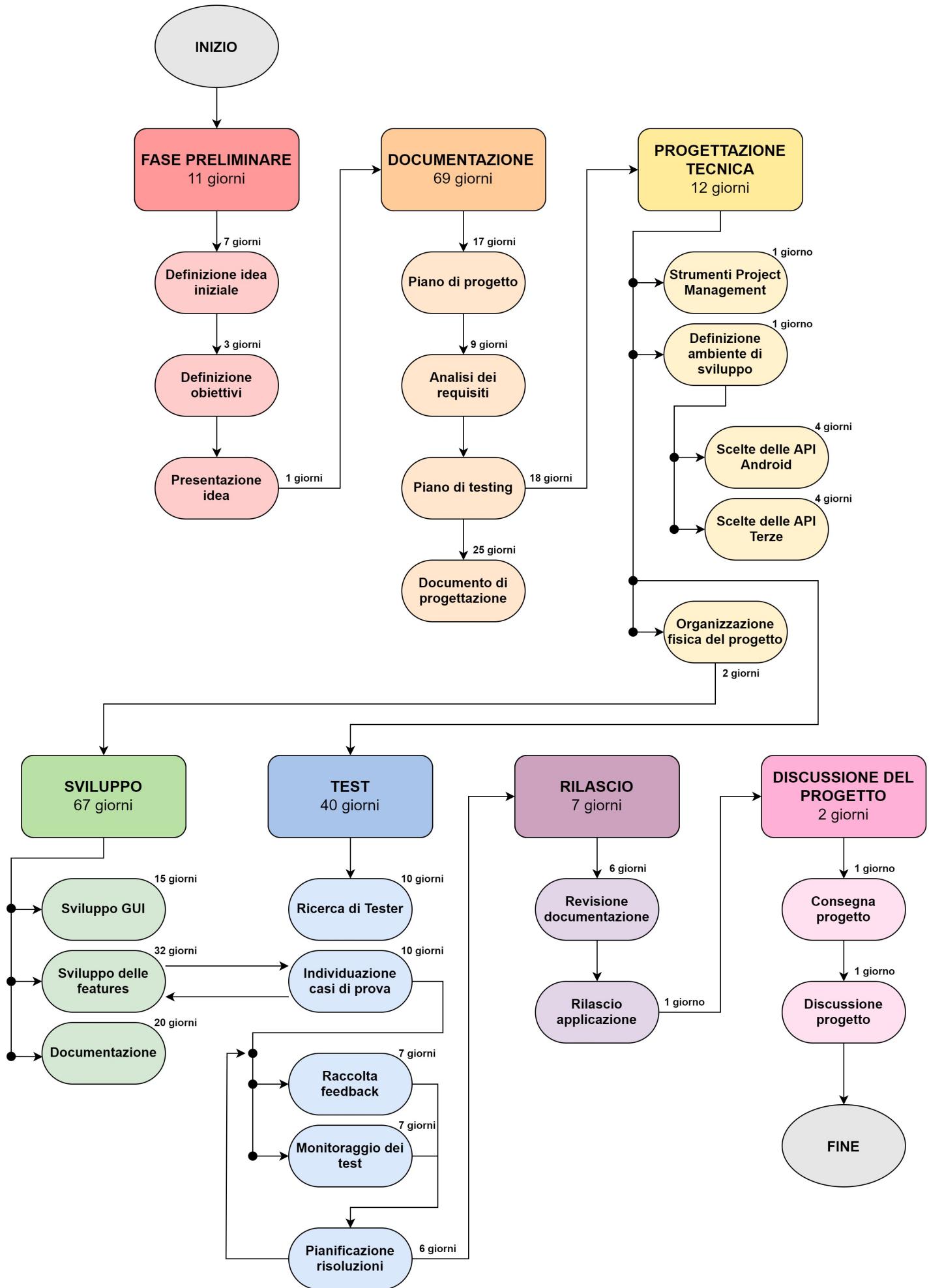
- [Diagramma su Diagrams.net interattivo](#) (Pagina 1)
- [Formato PDF \(vettoriale\)](#)
- [Immagine JPG \(300dpi\)](#)

Diagramma di Gantt:

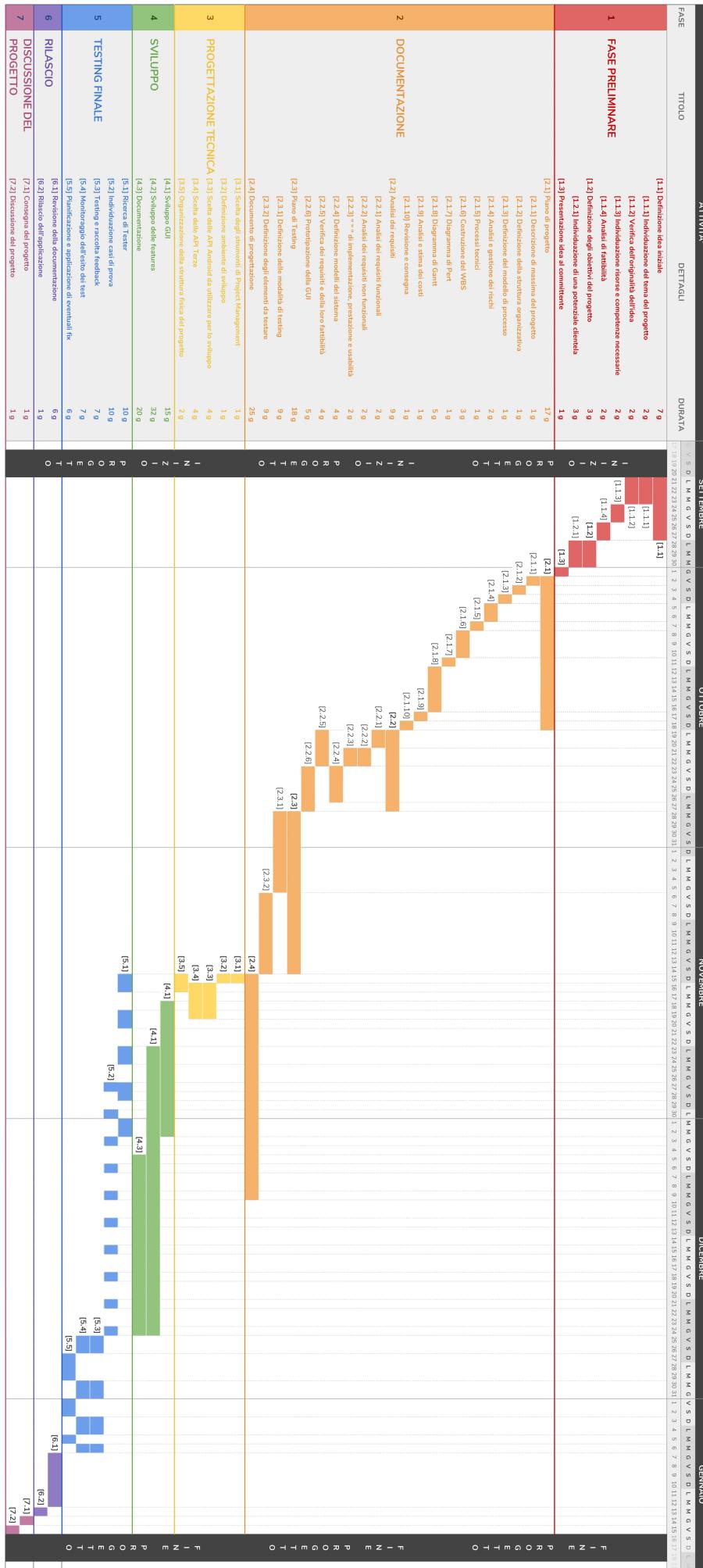
- [Foglio di calcolo Google Sheet interattivo](#)
- [Formato PDF \(vettoriale\)](#)

NB: Tali schemi **non** verranno aggiornati man mano che il progetto proseguirà, in quanto, per mantenere sotto controllo le tempistiche, faremo sempre riferimento al Software di Project Management "Trello".

[5.2.1] Diagramma di Pert (Figura 3)



[5.2.2] Diagramma di Gantt (Figura 4)



[5.3] Risorse Necessarie

Le **risorse** necessarie, utili alla riuscita del progetto, sono di **quattro tipi**:

- **Umane**: i componenti del gruppo che provvederanno allo sviluppo del progetto e persone terze aventi scopo di testare l'app;
- **Software**: tutti gli strumenti citati al [4.1], i quali sono tutti disponibili gratuitamente;
- **Hardware**: per lo sviluppo è necessario essere in possesso di computer e più dispositivi android;
- **Tempo**: per garantire una buona qualità del prodotto è necessaria una buona gestione del tempo da parte del singolo e del gruppo.

[5.4] Allocazione del Budget e delle Risorse

Analizzando il Diagramma di Gantt è possibile fornire una **stima dei costi** del progetto.

Ipotizzando di essere un'azienda con **4 dipendenti**, abbiamo assunto che un dipendente si occupa del nostro progetto per un tempo pari a **4 ore al giorno**.

Il salario orario che abbiamo previsto è di **12€ all'ora**.

Per favorire la leggibilità, **alleggiamo** nuovamente la tabella in vari formati:

- [**Foglio di calcolo Google Sheet interattivo**](#) (Pagina 2)
- [**Formato PDF \(vettoriale\)**](#)

Alla [pagina seguente](#) allegiamo la **tabella** riassuntiva dei **costi stimati** (Figura 5).

NB: I giorni totali sono calcolati sotto l'assunzione di esecuzione non parallela di alcuna attività. Ad esempio, due attività da 2 giorni ciascuna, eseguite in parallelo, occuperanno 4 giorni nella stima dei costi.

[5.5] Pianificazione

Le scadenze prefissate dal professore, dette **Deadlines**, sono quelle citate al punto [1.2].

In aggiunta, abbiamo fissato delle **Milestones** intermedie in modo da rendere più spezzettato lo sviluppo del progetto. In questo modo, è possibile **organizzare** al meglio le nostre **risorse**. Vedi punto [5.1].

N° ore lavorative:	4 ore/giorno
Salario:	12,00 €/h

* INDICA UNA CELLA CALCOLATA AUTOMATICAMENTE

DETTAGLI DEL PROGETTO					STIME DI TEMPO E COSTO	
TITOLO	DATA INIZIO	DATA FINE	TEAM	DURATA*	ORE / UOMO*	COSTO*
FASE PRELIMINARE	21/09/2020	01/10/2020	1,4	11 g	44 h	€ 960,00
[1.1] Definizione idea iniziale	21/09/2020	27/09/2020	1,3	7 g	28 h	€ 624,00
[1.1.1] Individuazione del tema del progetto	21/09/2020	23/09/2020	2	3 g	12 h	€ 288,00
[1.1.2] Verifica dell'originalità dell'idea	21/09/2020	23/09/2020	1	3 g	12 h	€ 144,00
[1.1.3] Individuazione risorse e competenze necessarie	24/09/2020	25/09/2020	1	2 g	8 h	€ 96,00
[1.1.4] Analisi di fattibilità	26/09/2020	27/09/2020	1	2 g	8 h	€ 96,00
[1.2] Definizione degli obiettivi del progetto	28/09/2020	30/09/2020	2	3 g	12 h	€ 288,00
[1.2.1] Individuazione di una potenziale clientela	28/09/2020	30/09/2020	2	3 g	12 h	€ 288,00
[1.3] Presentazione idea al committente	01/10/2020	01/10/2020	1	1 g	4 h	€ 48,00
DOCUMENTAZIONE	02/10/2020	09/12/2020	1,6	69 g	276 h	€ 6.528,00
[2.1] Piano di progetto	02/10/2020	18/10/2020	2,1	17 g	68 h	€ 2.256,00
[2.1.1] Descrizione di massima del progetto	02/10/2020	02/10/2020	1	1 g	4 h	€ 48,00
[2.1.2] Definizione della struttura organizzativa	03/10/2020	03/10/2020	1	1 g	4 h	€ 48,00
[2.1.3] Definizione del modello di processo	04/10/2020	04/10/2020	2	1 g	4 h	€ 96,00
[2.1.4] Analisi e gestione dei rischi	05/10/2020	06/10/2020	2	2 g	8 h	€ 192,00
[2.1.5] Processi tecnici	07/10/2020	07/10/2020	1	1 g	4 h	€ 48,00
[2.1.6] Costruzione del WBS	08/10/2020	10/10/2020	4	3 g	12 h	€ 576,00
[2.1.7] Diagramma di Pert	11/10/2020	11/10/2020	2	1 g	4 h	€ 96,00
[2.1.8] Diagramma di Gantt	12/10/2020	16/10/2020	4	5 g	20 h	€ 960,00
[2.1.9] Analisi e stima dei costi	17/10/2020	17/10/2020	3	1 g	4 h	€ 144,00
[2.1.10] Revisione e consegna	18/10/2020	18/10/2020	1	1 g	4 h	€ 48,00
[2.2] Analisi dei requisiti	19/10/2020	27/10/2020	2	9 g	36 h	€ 2.208,00
[2.2.1] Analisi dei requisiti funzionali	19/10/2020	20/10/2020	2	2 g	8 h	€ 192,00
[2.2.2] Analisi dei requisiti non funzionali	21/10/2020	22/10/2020	2	2 g	8 h	€ 192,00
[2.2.3] " " " di implementazione, prestazione e usabilità	21/10/2020	22/10/2020	1	2 g	8 h	€ 96,00
[2.2.4] Definizione modelli del sistema	23/10/2020	26/10/2020	3	4 g	16 h	€ 576,00
[2.2.5] Verifica dei requisiti e della loro fattibilità	19/10/2020	22/10/2020	1	4 g	16 h	€ 192,00
[2.2.6] Prototipazione della GUI	23/10/2020	27/10/2020	4	5 g	20 h	€ 960,00
[2.3] Piano di Testing	28/10/2020	14/11/2020	1	18 g	72 h	€ 864,00
[2.3.1] Definizione delle modalità di testing	28/10/2020	05/11/2020	1	9 g	36 h	€ 432,00
[2.3.2] Definizione degli elementi da testare	06/11/2020	14/11/2020	1	9 g	36 h	€ 432,00
[2.4] Documento di progettazione	15/11/2020	09/12/2020	1	25 g	100 h	€ 1.200,00
PROGETTAZIONE TECNICA	15/11/2020	19/11/2020	1,4	12 g	48 h	€ 960,00
[3.1] Scelta degli strumenti di Project Management	15/11/2020	15/11/2020	1	1 g	4 h	€ 48,00
[3.2] Definizione ambiente di sviluppo	15/11/2020	15/11/2020	1	1 g	4 h	€ 48,00
[3.3] Scelta delle API Android da utilizzare per lo sviluppo	16/11/2020	19/11/2020	2	4 g	16 h	€ 384,00
[3.4] Scelta delle API Terze	16/11/2020	19/11/2020	2	4 g	16 h	€ 384,00
[3.5] Organizzazione della struttura fisica del progetto	15/11/2020	16/11/2020	1	2 g	8 h	€ 96,00
Sviluppo	18/11/2020	24/12/2020	2,7	67 g	268 h	€ 8.688,00
[4.1] Sviluppo GUI	18/11/2020	02/12/2020	3	15 g	60 h	€ 2.160,00
[4.2] Sviluppo delle features	23/11/2020	24/12/2020	3	32 g	128 h	€ 4.608,00
[4.3] Documentazione	05/12/2020	24/12/2020	2	20 g	80 h	€ 1.920,00
TESTING FINALE	15/11/2020	06/01/2021	2,8	40 g	160 h	€ 4.992,00
[5.1] Ricerca di Tester	15/11/2020	02/12/2020	1	10 g	40 h	€ 480,00
[5.2] Individuazione casi di prova	27/11/2020	24/12/2020	2	10 g	40 h	€ 960,00
[5.3] Testing e raccolta feedback	25/12/2020	06/01/2021	4	7 g	28 h	€ 1.344,00
[5.4] Monitoraggio dell'esito dei test	25/12/2020	06/01/2021	4	7 g	28 h	€ 1.344,00
[5.5] Pianificazione e applicazione di eventuali fix	27/12/2020	05/01/2021	3	6 g	24 h	€ 864,00
RILASCIO	07/01/2021	13/01/2021	1,5	7 g	28 h	€ 624,00
[6.1] Revisione della documentazione	07/01/2021	12/01/2021	2	6 g	24 h	€ 576,00
[6.2] Rilascio dell'applicazione	13/01/2021	13/01/2021	1	1 g	4 h	€ 48,00
DISCUSSIONE DEL PROGETTO	14/01/2021	15/01/2021	2,5	2 g	8 h	€ 240,00
[7.1] Consegnna del progetto	14/01/2021	14/01/2021	1	1 g	4 h	€ 48,00
[7.2] Discussione del progetto	15/01/2021	15/01/2021	4	1 g	4 h	€ 192,00

RIEPILOGO DEL PROGETTO*					STIME DI TEMPO E COSTO TOTALE*	
TITOLO	DATA INIZIO	DATA FINE	TEAM	DURATA	ORE / UOMO	COSTO
FASE PRELIMINARE	21/09/2020	01/10/2020	1,4	11 g	44 h	€ 960,00
DOCUMENTAZIONE	02/10/2020	09/12/2020	1,6	69 g	276 h	€ 6.528,00
PROGETTAZIONE TECNICA	15/11/2020	19/11/2020	1,4	12 g	48 h	€ 960,00
Sviluppo	18/11/2020	24/12/2020	2,7	67 g	268 h	€ 8.688,00
TESTING FINALE	15/11/2020	06/01/2021	2,8	40 g	160 h	€ 4.992,00
RILASCIO	07/01/2021	13/01/2021	1,5	7 g	28 h	€ 624,00
DISCUSSIONE DEL PROGETTO	14/01/2021	15/01/2021	2,5	2 g	8 h	€ 240,00
TOTALE*	21/09/2020	15/01/2021	2,0	208 g	832 h	€ 22.992,00

Figura 5: Analisi e Stima dei costi descritta al [5.4]

NOTA RIGUARDANTE LA FASE DI TESTING

Come pianificato e mostrato nella WBS, il Team ha previsto una **fase di testing pubblica**. In particolare, la **ricerca dei tester** è avvenuta utilizzando il [seguente questionario](#).

La fase di testing vera e propria, inizialmente **prevista** per le **date comprese tra il 25/12/2020 e il 06/01/2021**, **non sarà effettuabile**, date le normative vigenti **causa Covid-19**.

Speriamo di **testare l'applicazione non appena possibile**, a patto di aver **tempo sufficiente** per la correzione di eventuali bug riscontrati.

In ogni caso, il **Team provvederà**, in modo limitato, a **testare l'applicazione in privato**, facendosi aiutare dalle rispettive famiglie.