



Università
Ca'Foscari
Venezia

Progetto di ingegneria del software 2023/2024
Piano di progetto per GaSmart
Versione 2.0

Gruppo Argenta

15 gennaio 2024

Document informations

| | |
|-------------------------|--|
| Deliverable | Piano di progetto |
| Data di consegna | 15/01/2024 |
| Team leader | Donald Gera - 892604@stud.unive.it |
| Team members | Daniel Andrei Bercu - 891470@stud.unive.it Mattia Schiavon - 890993@stud.unive.it Marco Netti - 892399@stud.unive.it Sebastiano Sartor - 891825@stud.unive.it |

Document history

| Version | Issue Date | Stage | Changes | Contributors |
|---------|------------|-------|-----------------------------------|---|
| 1.0 | 17/10/2023 | Draft | Creazione piano di progetto | Mattia Schiavon, Marco Netti, Sebastiano Sartor, Daniel Andrei Bercu, Donald Gera |
| 2.0 | 15/01/2024 | Final | Allineamento della documentazione | Mattia Schiavon |

Contents

| | | |
|----------|--|-----------|
| 1 | Introduzione | 3 |
| 1.1 | Overview del progetto | 3 |
| 1.1.1 | Funzionalità principali | 3 |
| 1.2 | Deliverables del progetto | 3 |
| 1.3 | Evoluzione | 3 |
| 1.4 | Materiale di riferimento | 4 |
| 1.4.1 | Definizioni e abbreviazioni | 4 |
| 2 | Organizzazione del progetto | 5 |
| 2.1 | Modello del processo | 5 |
| 2.2 | Struttura organizzativa | 5 |
| 2.3 | Interfacce organizzative | 6 |
| 2.4 | Responsabilità di progetto | 6 |
| 3 | Processi gestionali | 7 |
| 3.1 | Obiettivi e priorità | 7 |
| 3.1.1 | Obiettivi | 7 |
| 3.1.2 | Priorità | 7 |
| 3.2 | Assunzioni, dipendenze, vincoli | 7 |
| 3.3 | Gestione dei rischi | 8 |
| 3.3.1 | Tabella dei rischi | 10 |
| 3.3.2 | Tabella di prevenzione e monitoraggio dei rischi | 11 |
| 3.3.3 | Modello dei rischi | 11 |
| 3.4 | Meccanismi di monitoraggio e di controllo | 12 |
| 3.5 | Pianificazione dello staff | 12 |
| 4 | Processi tecnici | 13 |
| 4.1 | Metodi, strumenti, tecniche | 13 |
| 4.1.1 | Ambiente di sviluppo | 13 |
| 4.1.2 | Metodi di sviluppo | 13 |
| 4.1.3 | Linguaggio di programmazione e framework | 13 |
| 4.2 | Documentazione del software | 14 |
| 4.3 | Funzionalità di supporto al progetto | 14 |
| 5 | Pianificazione del lavoro, delle risorse umane e budget | 15 |
| 5.1 | WBS - Work Breakdown Structure | 15 |
| 5.2 | Dipendenze | 18 |
| 5.3 | Risorse Necessarie | 20 |
| 5.4 | Allocazione del Budget e delle Risorse | 20 |
| 5.5 | Pianificazione | 21 |

1 Introduzione

1.1 Overview del progetto

Il gruppo Argenta tramite il progetto **GaSmart** vuole fornire ad una variegata fascia di utenti un'applicazione Android che permetta loro di avere più informazioni durante la scelta del distributore di carburante, focalizzandosi su risparmio e impatto ambientale.

1.1.1 Funzionalità principali

- **Ricerca del veicolo:** l'applicazione permette all'utente di ricercare il proprio veicolo all'interno di un dataset, se non è presente viene fornita la possibilità di inserire i valori manualmente. L'inserimento di valori personalizzati per un veicolo risulta inoltre utile nel caso l'utente sia a conoscenza di una variazione dei consumi del proprio veicolo.
- **Visualizzazione dei distributori secondo una preferenza data:** durante la configurazione dell'applicazione all'utente verrà chiesto di indicare, oltre alle specifiche sul veicolo e sul rifornimento medio, le seguenti informazioni ai fini di migliorare la sua esperienza:
 - raggio massimo dalla posizione attuale a cui si può trovare un distributore
 - se preferisce massimizzare il risparmio, minimizzare l'inquinamento o una via di mezzo (valore di default)

L'applicazione fornirà poi un lista di distributori ordinati in base alle preferenze indicate dall'utente, per ogni distributore verranno indicati i dati chiave.

- **Navigazione:** l'utente può selezionare uno dei distributori e scegliere di far iniziare la navigazione in una delle app che lo consentono.
- **Statistiche:** ogni volta che l'utente effettua un rifornimento può inserire le informazioni sulla spesa, dati questi che verranno poi aggregati e presentati all'utente per renderlo consapevole dei suoi consumi di carburante.

1.2 Deliverables del progetto

| Nome deliverable | Data consegna | Luogo di consegna |
|---|---------------|-------------------|
| Piano di progetto | 17/10/2023 | Moodle del corso |
| Documento dei requisiti | 31/10/2023 | Moodle del corso |
| Piano di testing | 14/11/2023 | Moodle del corso |
| Documento di progettazione | 28/11/2023 | Moodle del corso |
| Versione 1.0 del codice sorgente | 15/12/2023 | Moodle del corso |
| Versione 1.1 del codice e allineamento della documentazione | 15/01/2023 | Moodle del corso |

1.3 Evoluzione

Dato che siamo ancora nelle fasi iniziali i piani per eventuali cambiamenti rispetto all'idea iniziale non sono ancora ben definiti. Ci aspettiamo di poter far evolvere e far aumentare il potenziale del progetto servendoci delle opinioni dei potenziali utenti che andremo ad intervistare. Inoltre abbiamo in piano di utilizzare attivamente la nostra applicazione in prima persona, così da poter individuare eventuali punti deboli.

Guardando invece al codebase, Flutter e l'utilizzo di librerie costantemente aggiornate ci permettono di assicurare che l'applicazione possa progredire senza grandi problemi durante l'evoluzione del sistema Android, inoltre in futuro ci sarà possibile far evolvere il progetto in un'ottica multiplatform senza difficoltà.

1.4 Materiale di riferimento

- [Materiale fornito nel corso di Ingegneria del software](#)
- [Documentazione ufficiale di Flutter](#)
- [Documentazione ufficiale di Dart](#)
- [Documentazione ufficiale di Android](#)
- [Documentazione di Google Maps Platform](#)
- [StackOverflow](#) - Per eventuali dubbi o problematiche comuni
- [Medium](#) - Vengono utilizzati vari articoli per arricchire il progetto, soprattutto dal punto di vista del codice

1.4.1 Definizioni e abbreviazioni

- **Android:** è un sistema operativo per dispositivi mobili sviluppato da Google, ad oggi è presente in una vasta gamma di dispositivi tra i quali smartphone (principalmente), televisori, automobili e orologi. [Fonte](#)
- **App (applicazione mobile):** in informatica un'applicazione mobile è un'applicazione dedicata specificatamente a dispositivi mobili, tipicamente è realizzata con un occhio di riguardo al consumo di risorse. [Fonte](#)
- **Google Maps:** è un servizio internet geografico sviluppato da Google che consente la ricerca e la visualizzazione di carte geografiche di buona parte della Terra. [Fonte](#)
- **GitHub:** è un servizio di hosting pensato principalmente per progetti software. [Fonte](#)
- **Trello:** è un software gestionale in stile Kanban basato sul web. [Fonte](#)
- **Kanban:** in ambito di sviluppo software si fa riferimento a Kanban come una metodologia di sviluppo software, in generale si procede come segue: i singoli compiti vengono considerati delle piccole fasi e portati a termine in sequenza, uno alla volta. [Fonte](#)
- **Android Studio:** è un ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaforma Android. [Fonte](#)
- **API:** con **A**pplication **P**rogramming **I**nterface si indica un insieme di procedure atte a risolvere uno specifico problema di comunicazione tra diversi computer o tra diversi software o tra diversi componenti di software. [Fonte](#)
- **Google Play:** Serve come app store ufficiale, per il sistema operativo Android, consentendo agli utenti di navigare e scaricare applicazioni sviluppate con il kit di sviluppo software Android (SDK) e pubblicate tramite Google. [Fonte](#)
- **F-Droid:** è un client per dispositivi Android che aggrega il catalogo delle applicazioni interrogando periodicamente i depositi di software (repository) configurati, tipicamente quello ufficiale gestito dalla comunità di F-Droid, che contiene esclusivamente software libero. [Fonte](#)
- **Flutter:** è un framework open-source creato da Google per la creazione di interfacce native per iOS, Android, Linux, macOS e Windows oltre ad essere il metodo principale per la creazione di applicazioni per Google Fuchsia. [Fonte](#)
- **Dart:** è un linguaggio di programmazione sviluppato da Google, il suo scopo è quello di sostituire JavaScript come lingua franca per lo sviluppo web sulla piattaforma "web aperto". Il compilatore Dart permette di scrivere programmi sia per il web che per desktop e server. [Fonte](#)

2 Organizzazione del progetto

2.1 Modello del processo

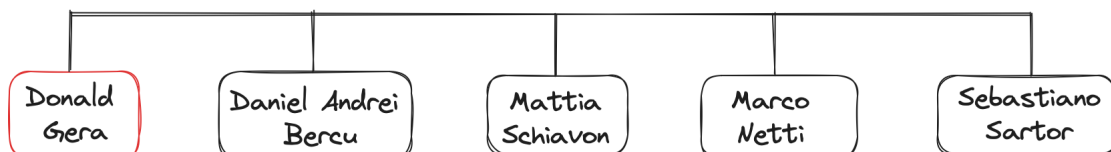
Nell'ambito dello sviluppo dell'app in questione, il processo di realizzazione adotterà il modello Agile come base. In particolare rispettando i principi descritti nel manifesto

- Le persone e le interazioni sono più importanti dei processi e degli strumenti:
In questo contesto, mettiamo un'enfasi particolare sulle relazioni e la comunicazione tra i membri del team. Riteniamo che queste siano la risorsa più preziosa per il successo del progetto. Favoriamo un'apertura al dialogo e la collaborazione continua tra i membri del team, riconoscendo che la comunicazione efficace è essenziale per il raggiungimento degli obiettivi.
- È più importante avere software funzionante che documentazione:
La nostra filosofia è quella di rilasciare nuove versioni dell'applicazione a intervalli regolari, concentrandoci sulla creazione di software funzionante e mantenendo il codice semplice ed efficiente. Questo non significa trascurare la documentazione, ma piuttosto essere più consapevoli su ciò su cui si investe tempo ed energie.
- La collaborazione diretta e continua con i nostri utenti è un elemento cruciale:
Abbiamo identificato un insieme di potenziali utilizzatori della app da coinvolgere in modo attivo per identificare le funzionalità a cui dare priorità e le caratteristiche dei servizi che possono meglio incontrare bisogni effettivi ed aspettative realizzabili. Il nostro target principale sono gli utenti con un'età compresa tra i 20 e i 50 anni, cercheremo di intervistare un plateau di persone il più eterogenea possibile, con un certo focus per gli utenti non nativi digitali.
- Siamo pronti ad adattarci ai cambiamenti:
La nostra squadra è aperta e flessibile, pronta a gestire cambiamenti e modifiche nel progetto quando necessario.

La decisione di adottare il modello Agile ci consentirà inoltre di assicurare una scrittura del codice più sicura, una visione più chiara e completa del processo di sviluppo e la possibilità di apportare modifiche in corso d'opera senza dover affrontare rischi inutili, preservando al contempo le risorse e il tempo per rispondere alle varie esigenze degli utenti. In aggiunta, il nostro team si è impegnato e continuerà a condurre 2 incontri settimanali. Questi incontri serviranno non solo per discutere lo stato di avanzamento del progetto, ma anche per affrontare eventuali sfide o difficoltà emergenti e per valutare possibili aggiustamenti garantendo così una gestione più efficiente e una migliore comunicazione all'interno del team.

2.2 Struttura organizzativa

Il team ha adottato un approccio democratico per la presa di decisioni, che include l'uso di sondaggi per raggiungere un consenso. Quando si rende necessario proporre modifiche al progetto, stabiliamo una data entro la quale ciascun membro del gruppo deve presentare la propria idea o esporre l'opinione sul tema in questione. Successivamente, avviamo un sondaggio per raccogliere i pareri di tutti i membri del team. Inoltre, cerchiamo di assegnare compiti in modo mirato, sfruttando le competenze e le conoscenze pregresse di ciascun membro del gruppo, al fine di massimizzare l'efficienza e la qualità del lavoro svolto. Sondaggi e discussioni avvengono principalmente su WhatsApp, per incontri più intensi preferiamo l'utilizzo di Discord in quanto ci permette di condividere materiale e far visualizzare lo schermo del proprio pc in tempo reale agli altri membri del gruppo.



2.3 Interfacce organizzative

Nel corso dello sviluppo del progetto ci troveremo spesso nella situazione di dover comunicare con parti esterne al nostro team. In particolare, prevediamo di contattare il Professor A. Cortesi nel caso di dubbi o domande, al fine di ottenere chiarezza e consulenza in merito ai dettagli tecnici e organizzativi del progetto. Faremo affidamento sulla sua esperienza e sui suoi consigli per garantire un risultato finale di alta qualità. Inoltre, sarà fondamentale che tutti i membri del gruppo seguano attentamente le lezioni e si impegnino attivamente nel processo di apprendimento. Questo ci permetterà di applicare al meglio i consigli e le indicazioni fornite. Per quanto riguarda le questioni di natura tecnica, faremo affidamento sul Professor A. Spanò e ci impegneremo a risolvere autonomamente eventuali dubbi o problemi, cercando soluzioni e sfruttando le risorse a nostra disposizione, tra cui la documentazione e le competenze dei membri del team. Inoltre, come menzionato precedentemente prevediamo di collaborare con i futuri tester del nostro prodotto GaSmart per raccogliere feedback da loro e apportare eventuali modifiche in base alle sfide o ai problemi che emergono durante il processo di testing.

2.4 Responsabilità di progetto

Il nostro approccio prevede che il team operi in modo uniforme e collaborativo, evitando una divisione netta in team distinti così da rendere inoltre ogni membro del team un protagonista del progetto. Questa flessibilità ci consente di adattarci alle necessità settimanali del progetto. Si delineano dei ruoli specifici per i vari membri del team che possono variare nel corso del tempo così da permettere di concentrarsi su differenti aspetti del progetto:

- Sebastiano Sartor si concentrerà sull'applicazione e sulle questioni relative alla sicurezza
- Mattia Schiavon si concentrerà invece sullo sviluppo del server che dovrà erogare le API utilizzare dall'applicazione
- Donald Gera, Daniel Andrei Bercu e Marco Netti si concentreranno maggiormente sulla stesura e il mantenimento dei deliverable

Si sottolinea come ogni componente del team prenderà in realtà parte a tutti gli aspetti dello sviluppo del progetto. La decisione sui task e sulle attività da svolgere è presa di comune accordo da tutti i membri del gruppo, promuovendo la collaborazione e la condivisione delle responsabilità. Inoltre, la struttura del team è suscettibile di modifiche per adattarsi alle esigenze specifiche di ciascuna fase del progetto. Questo approccio ci consente di massimizzare l'efficienza e la qualità del lavoro.

3 Processi gestionali

3.1 Obiettivi e priorità

I nostri obiettivi e priorità delineano il percorso da seguire nel corso dello sviluppo del progetto. Sono fondamentali in quanto ci guidano verso la realizzazione di un prodotto che funzioni in modo ottimale e che soddisfi le aspettative stabilite.

Tenendo costantemente presenti questi due elementi durante l'intero sviluppo, assicuriamo che il risultato finale sia all'altezza delle nostre aspettative e dei requisiti del progetto.

3.1.1 Obiettivi

Gli obiettivi elencati sotto corrispondono, sostanzialmente, alle funzionalità chiave che intendiamo implementare nell'applicazione. Ciò significa che ognuno di questi obiettivi non rappresenta solo un punto di riferimento generico, ma si traduce concretamente in un servizio che vogliamo che l'app offra agli utenti.

- **Ottimizzazione della scelta del distributore**
- **Elevata usabilità e personalizzazione**
- **Navigazione efficiente verso il distributore scelto**
- **Configurazione intuitiva delle preferenze**
- **Presentazione chiara e organizzata dei distributori**
- **Tracciamento e analisi delle scelte dell'utente**

3.1.2 Priorità

- **Rispettare le scadenze con precisione:** assicurare che tutte le scadenze siano rigorosamente rispettate è essenziale per garantire una gestione organizzata e la consegna puntuale di tutti i deliverable. Questo contribuirà a mantenere l'andamento del progetto in linea con le aspettative e a evitare possibili ritardi.
- **Considerare la diversità degli utenti:** riconoscere l'eterogeneità degli utenti è fondamentale. Dovremo creare un sistema che tenga conto dei punti critici per le varie categorie di utenti, ad esempio fornendo soluzioni accessibili e intuitive a livello grafico, soprattutto per coloro che potrebbero avere esigenze specifiche come gli anziani.
- **Centrare l'utilità per l'utente finale:** durante il processo di sviluppo, sarà cruciale prendere in considerazione le esigenze e le preferenze ipotetiche degli utenti finali del prodotto. Questo ci consentirà di progettare un'app che soddisfi al meglio le loro necessità, garantendo un'esperienza utente soddisfacente e funzionale.
- **Offrire un'esperienza utente coinvolgente:** non basta che il prodotto sia funzionale. È altrettanto importante che l'esperienza utente sia piacevole e coinvolgente. Ciò significa che l'applicazione dovrà essere intuitiva nell'uso, o un design che faciliti la navigazione e una presentazione visiva che favorisca l'interazione.
- **Conseguire in concreto gli obiettivi prefissati:** dobbiamo impegnarci a completare con successo l'insieme dei propositi che abbiamo definito inizialmente. Questo richiederà una pianificazione accurata, una gestione attenta delle risorse e un costante monitoraggio del progresso.

3.2 Assunzioni, dipendenze, vincoli

- **Assunzioni:** Dopo le lezioni di Android del Prof. Spanò, ci aspettiamo che tutti i membri del gruppo abbiano acquisito una buona comprensione dello sviluppo su Android Studio. Tuttavia, è probabile che siano necessari ulteriori approfondimenti in quanto il processo di sviluppo è un percorso continuo. Inoltre, supponiamo che ciascun membro del gruppo abbia delle basi solide con la costruzione di un database. Questo è cruciale per garantire la gestione

efficace delle informazioni all'interno dell'applicazione.

Ci aspettiamo un impegno significativo da parte di ciascun membro del gruppo per portare a termine il lavoro. Ogni contributo e partecipazione attiva sono fondamentali per raggiungere gli obiettivi che ci siamo prestabiliti per il progetto.

- **Dipendenze:** L'uso delle API di Google Maps sarà fondamentale per l'implementazione di funzionalità sulla geolocalizzazione, consentendo agli utenti di interagire con mappe dinamiche all'interno dell'applicazione. Inoltre, acquisiremo dati riguardanti i prezzi e le posizioni dei distributori di benzina da una dataset fornito da fonti governative, fornendo le informazioni necessarie per determinare la stazione di servizio più conveniente in base a criteri di distanza e prezzo.

L'integrazione sinergica di queste dipendenze sarà di fondamentale importanza per la realizzazione di un'applicazione completa e funzionale, che offrirà agli utenti una soluzione pratica e intuitiva per individuare la stazione di servizio più vantaggioso.

- **Vincoli:** In primo luogo, è fondamentale rispettare le scadenze stabilite dal professore. Questo contribuirà a mantenere un progresso costante e a garantire che il progetto proceda secondo il piano stabilito. Inoltre, il progetto deve mirare alla realizzazione di un'applicazione completa e sicura. Questo significa che siamo tenuti a considerare e implementare tutte le misure necessarie per garantire la sicurezza delle informazioni. Infine, l'esperienza utente è molto importante. L'applicazione deve offrire un'interfaccia intuitiva e una navigazione fluida, in modo che gli utenti possano sfruttare appieno le funzionalità offerte. Sarà essenziale garantire che il controllo del vicinato sia intuitivo e facilmente accessibile all'utente.

3.3 Gestione dei rischi

Nel corso dello sviluppo di un progetto che si estende su un determinato intervallo temporale, è essenziale considerare e gestire eventuali rischi che potrebbero insorgere, mantenendo una solida collaborazione di gruppo.

Al fine di garantire la tempestiva e sicura realizzazione del progetto, è essenziale individuare e gestire i possibili rischi. Di seguito, vengono presentati alcuni dei rischi che è importante evitare. Si raccomanda inoltre una periodica revisione e aggiornamento del piano per assicurare un progresso senza intoppi.

- **Rischi legati ai bug del codice:** durante lo sviluppo, è probabile incontrare errori che richiederanno un'attenta identificazione e correzione. Per mitigare questa eventualità, è cruciale pianificare un adeguato processo di testing, che consenta di individuare e risolvere i bug in modo tempestivo.
- **Difficoltà con il linguaggio di programmazione:** un altro fattore da considerare è la possibile difficoltà con il linguaggio di programmazione adottato per lo sviluppo. Considerando che solo un membro del gruppo ha esperienza con Flutter e Dart, è importante essere consapevoli della possibile difficoltà iniziale nell'approccio a questo nuovo linguaggio. Per mitigare questa situazione, ci impegniamo a dedicare del tempo allo studio e all'apprendimento di Flutter, cercando risorse e materiali che ci permettano di acquisire familiarità con esso.
- **Guasti hardware ai computer:** non possiamo escludere la possibilità di guasti hardware ai terminali utilizzati per lo sviluppo. In caso di un simile imprevisto, sarà necessario temporaneamente utilizzare una macchina alternativa, risolvendo successivamente il problema.
- **Guasti software ai computer:** anche i guasti software ai computer rappresentano un potenziale rischio, sebbene sia gestibile con le nostre competenze. Tuttavia, l'eventuale perdita di tempo per rimediare al guasto potrebbe essere significativa.
- **Perdita del lavoro svolto:** la perdita di lavoro già svolto costituisce un rischio critico, potenzialmente dannoso per l'intero progetto. È pertanto vitale mantenere diverse versioni di backup utilizzando i servizi come Google Drive e GitHub, assicurandoci di preservare in modo sicuro il progresso del lavoro.

-
- **Problematiche relative all'archiviazione nel cloud:** l'utilizzo limitato del servizio cloud nel nostro progetto per l'archiviazione di documenti e codice sorgente riduce i rischi potenziali di perdere tali dati. Tuttavia, è importante considerare possibili problemi come l'accesso limitato ai dati in caso di interruzioni del servizio o problemi di connessione. Inoltre, è essenziale monitorare eventuali modifiche in termini di servizio o politiche di archiviazione per adattare adeguatamente la gestione della documentazione e del codice sorgente dell'applicazione.
 - **Scarsa sinergia nel team:** come abbiamo detto nei punti precedenti, è fondamentale garantire una collaborazione efficace tra i membri del gruppo durante l'intero sviluppo del progetto. Tuttavia, potrebbero sorgere conflitti interni tra alcuni componenti, i quali potrebbero causare ritardi inaspettati, variazioni nei piani, e in situazioni estreme, potrebbero addirittura portare all'uscita di alcuni partecipanti dal gruppo.
 - **Pianificazione inadeguata:** a causa della limitata esperienza dei membri del gruppo nella gestione e organizzazione di progetti, c'è il rischio di una pianificazione non ottimale dei vari compiti assegnati. Questo potrebbe portare una perdita di tempo dovuta alla necessità di rivedere documenti e codice. È importante considerare che una pianificazione inefficace potrebbe influire negativamente sul progresso complessivo del progetto, portando a ritardi e difficoltà nel raggiungere gli obiettivi stabiliti. Pertanto, è fondamentale adottare strategie di pianificazione più solide e monitorare attentamente l'avanzamento dei compiti per evitare questo potenziale rischio.
 - **Problemi di salute:** è importante considerare i possibili problemi di salute che potrebbero influire sullo sviluppo del progetto. In questo periodo, l'incidenza delle influenze è più elevata, e potrebbero rallentare i progressi. In caso di necessità, è essenziale affrontare questa eventualità supportandosi reciprocamente, assicurandosi che il proseguimento dei lavori non ne risenta in modo significativo.

3.3.1 Tabella dei rischi

- **Legenda probabilità dei rischi**

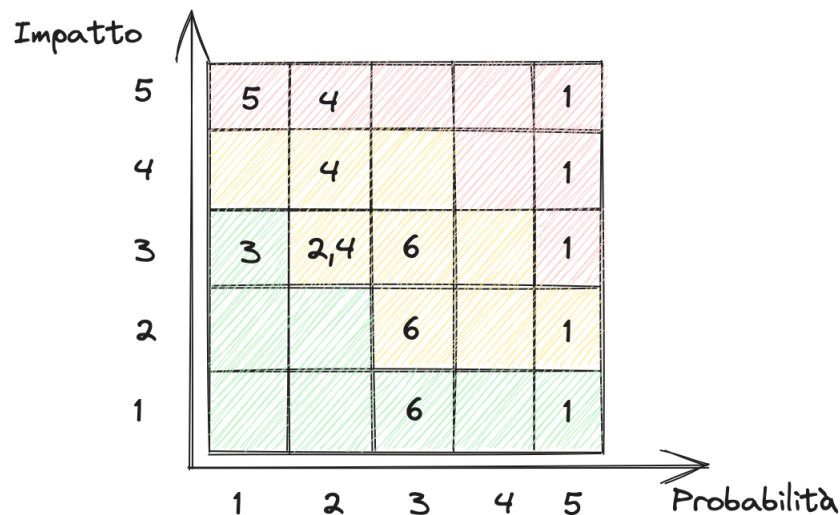
- 1 = Probabilità molto bassa;
- 2 = Probabilità bassa;
- 3 = Probabilità media;
- 4 = Probabilità alta;
- 5 = Probabilità altissima.

| ID | RISCHI | CATEGORIA | PROB. | IMPATTO | AZIONE |
|----|--------------------------------------|-----------|-------|---------|---|
| 1 | Bugs | Sviluppo | 5 | 1-5 | Revisione e modifiche al codice se necessario. |
| 2 | Difficoltà con il linguaggio | Sviluppo | 2 | 3 | Rivedere le regole di programmazione e cercare ulteriori informazioni online. |
| 3 | Guasto hardware | Materiale | 1 | 3 | Acquisire componenti di ricambio o, in casi critici, ottenere una macchina di sostituzione. |
| 4 | Guasto software | Sviluppo | 1 | 3-5 | Risolvere l'errore online o, in caso di emergenza, formattare la macchina. |
| 5 | Perdita del lavoro svolto | Sviluppo | 1 | 5 | Recuperare la copia dal cloud e ripristinarla come backup, se disponibile. |
| 6 | Problematiche di archiviazione cloud | Sviluppo | 2 | 3 | Contattare il supporto tecnico del provider. |
| 7 | Scarsa sinergia nel team | Sviluppo | 1 | 4 | Organizzare incontri team-building e promuovere la comunicazione aperta per rafforzare il rapporto |
| 8 | Pianificazione inadeguata | Sviluppo | 3 | 4 | Organizzare una riunione di team per discutere e chiarire gli obiettivi, le assegnazioni e le scadenze. |
| 9 | Problemi di salute | Sviluppo | 4 | 1-3 | Collaborare con il team e suddividere il lavoro in caso di malattia. |

3.3.2 Tabella di prevenzione e monitoraggio dei rischi

| ID | RISCHI | CAT. | PREVENZIONE | MONITOR. |
|----|--------------------------------------|-----------|--|---|
| 1 | Bugs | Sviluppo | Fornire particolare cura nella stesura del codice. | Verifica degli errori nel codice redatto. |
| 2 | Difficoltà con il linguaggio | Sviluppo | Iniziare l'apprendimento del linguaggio da zero. | Verifica e in caso modifica codice. |
| 3 | Guasto hardware | Materiale | Evitare di esporre i propri dispositivi a situazioni pericolose. | Controllo dell'integrità dell'hardware. |
| 4 | Guasto software | Sviluppo | Valutare attentamente cosa e come si installa sul proprio dispositivo. | Controllo eventuali segnali software anomali. |
| 5 | Perdita del lavoro svolto | Sviluppo | Utilizzare archiviazione multipla e servizi cloud. | Controllare che il lavoro sia stato sincronizzato online. |
| 6 | Problematiche di archiviazione cloud | Sviluppo | Eseguire backup regolari e garantire una connessione internet affidabile. | Controllare lo stato dei server dei servizi di cloud utilizzati. |
| 7 | Scarsa sinergia nel team | Sviluppo | Comunicazione e chiare assegnazioni di compiti. | Utilizzare strumenti di gestione del progetto e condurre regolari incontri di team. |
| 8 | Pianificazione inadeguata | Sviluppo | Definire chiaramente gli obiettivi, suddividere i compiti e ruoli in modo dettagliato e congruo con le competenze di ciascun membro. | Tenere riunioni di gruppo regolari e utilizzare il diagramma di Gantt per visualizzare le scadenze. |
| 9 | Problemi di salute | Sviluppo | Evitare situazioni di possibile contagio fisico. | Controllo di eventuali sintomi. |

3.3.3 Modello dei rischi



3.4 Meccanismi di monitoraggio e di controllo

Nel corso dello sviluppo del progetto, adotteremo un approccio collaborativo all'interno del team. Questo significa che il lavoro sarà costantemente sotto la supervisione attenta di tutti i membri, consentendo un controllo incrociato sulle varie componenti, sia documentazione che sia codice sorgente. Per ottimizzare alcune procedure di implementazione più semplici, assegneremo specifici compiti a diversi membri del team. Questa pratica intensificherà i controlli su porzioni di codice più specifiche, contribuendo a migliorare significativamente l'efficienza del processo di sviluppo.

Per facilitare la comunicazione e il confronto tra i membri del team, faremo uso di canali come Discord e WhatsApp. Questi strumenti ci consentiranno di mantenere una comunicazione diretta e immediata, permettendo di risolvere qualsiasi questione o dubbio che possa emergere durante il processo di sviluppo.

3.5 Pianificazione dello staff

Per portare a termine con successo il progetto, è essenziale che i membri siano proficienti nel linguaggio di programmazione Kotlin e acquisiscano una solida familiarità con Android Studio. Inoltre, sarà importante che tutti i membri siano in grado di utilizzare GitHub per gestire il controllo delle versioni e gestire i diversi branch necessari per l'implementazione e il testing delle varie caratteristiche del progetto. Questo garantirà un processo di sviluppo organizzato e ben strutturato, consentendo una gestione efficiente delle modifiche e una documentazione completa di tutti i progressi del progetto.

4 Processi tecnici

4.1 Metodi, strumenti, tecniche

4.1.1 Ambiente di sviluppo

Per l'implementazione di questo progetto, abbiamo identificato la necessità di utilizzare un ambiente di sviluppo specifico per le applicazioni Android. A tal fine, abbiamo scelto Android Studio, disponibile su qualsiasi sistema operativo desktop, che offre la possibilità di virtualizzare un dispositivo Android per testare l'applicazione. Ciò semplifica il processo di sviluppo anche per i membri del team che non dispongono un dispositivo Android nativo. Tuttavia, abbiamo anche ritenuto essenziale condurre test su dispositivi fisici per garantire la validità delle funzionalità e risolvere eventuali discrepanze tra l'ambiente virtuale e le implementazioni reali.

4.1.2 Metodi di sviluppo

Abbiamo adottato un approccio metodologico che prevede la suddivisione del progetto in task distinte. Successivamente, i membri del team saranno organizzati in gruppi a cui sarà assegnato un task specifico. Ogni settimana, stabiliremo scadenze intermedie per garantire il processo costante. Nel caso in cui un gruppo completi rapidamente, il proprio task mentre un altro incontra delle difficoltà, promuoveremo la collaborazione tra i gruppi per risolvere le problematiche in tempi ridotti.

4.1.3 Linguaggio di programmazione e framework

Abbiamo deciso di basare il progetto sul framework Flutter, andando quindi ad utilizzare come linguaggio di programmazione principale Dart. Questa decisione è stata motivata principalmente dalla possibilità di sviluppo cross-platform (una richiesta avanzata più volte dalle persone che abbiamo intervistato) e dalle alte prestazioni offerte da Dart.

Flutter inoltre facilita lo sviluppo dell'interfaccia utente fornendo widget personalizzabili e una libreria grafica flessibile. È presente l'hot-reload che permette modifiche istantanee, ottimizzando il processo di sviluppo.

Va precisato che Flutter gode di pieno supporto da parte di Google e ha una comunità molto attiva alle spalle, questo consente a noi sviluppatori di avere accesso a risorse di alta qualità, documentazione e supporto tecnico affidabile.

4.2 Documentazione del software

Durante il corso dello sviluppo del progetto, verranno redatti e aggiornati diversi documenti cruciali, tra cui:

- Piano di Progetto;
- Documento di Analisi;
- Documento di Analisi Specifica;
- Piano di Testing;
- Documento di Progettazione.

Tutti questi documenti saranno consegnati nella loro versione finale entro il 15/01/2024.

4.3 Funzionalità di supporto al progetto

Il nostro obiettivo primario è sviluppare un progetto con codice di alta qualità. A tale scopo, ci impegniamo a mantenere costantemente aggiornata la documentazione, sia riguardante il codice che il progetto in sé. Inoltre, faremo ampio uso di design patterns per garantire la creazione di un'applicazione con codice riutilizzabile e di facile comprensione. Prestiamo particolare alla scrittura efficiente in termini di tempo di esecuzione e utilizzo delle risorse, poiché l'applicazione sarà destinata ad eseguire su dispositivi con capacità di calcolo limitata. Effettueremo ciclicamente controlli sul codice e testeremo l'applicazione per rilevare potenziali bug o problematiche che potrebbero influire sull'esperienza dell'utente.

Per supportare il nostro processo di sviluppo, il team ha scelto di utilizzare una serie di strumenti, tra cui GitHub per il versioning e la collaborazione, Google Drive per la condivisione dei documenti e delle risorse, Android Studio come ambiente di sviluppo della documentazione, WhatsApp e Discord per la comunicazione del team, e Trello come strumento di gestione del progetto.

5 Pianificazione del lavoro, delle risorse umane e budget

5.1 WBS - Work Breakdown Structure

Con l'obiettivo di dividere il carico del lavoro tra i membri del team, è stata compiuta un'analisi del progetto a partire dalla sua interezza per arrivare alla distinzione delle singole attività, funzioni e *task* che lo compongono.

In questo modo, insieme alle indicazioni e alle scadenze date dal professor Cortesi, siamo riusciti a delineare le seguenti funzioni:

F1. Training

La scelta di utilizzare il framework Flutter per lo sviluppo del progetto rappresenta una novità per parte del team: in parallelo alle fasi di progettazione e sviluppo, ogni membro approfondirà lo studio del linguaggio scelto.

F2. Elaborazione documentazione

Come requisito del corso, ogni fase del progetto deve essere accompagnata dalla stesura di un documento che tenga conto dell'andamento del progetto. Di seguito sono elencati i documenti da consegnare:

- **Piano di Progetto:** Il Piano di Progetto, ovvero il presente documento, esplica le fasi di sviluppo e come esse verranno affrontate.
- **Documento dei Requisiti:** Il Documento dei Requisiti espone al committente i requisiti necessari per lo sviluppo del progetto.
- **Piano di Testing:** Il Piano di Testing riguarda l'ideazione e la realizzazione dei test per verificare il corretto funzionamento dell'applicazione, con le relative date.
- **Documento di Progettazione:** Il Documento di Progettazione affronta i dettagli dell'applicazione, dall'architettura del sistema alla logica applicativa alla progettazione dell'interfaccia utente.

F3. Project Management

Funzione concorrente a tutte le altre, prevede che l'andamento del progetto venga coordinato dai membri tramite meeting settimanali in cui si discutono idee, problematiche e possibili soluzioni.

F4 Configuration Management

Affinché lo sviluppo possa procedere secondo ritmi scanditi e senza ritardi, ogni membro si dovrà assicurare che la propria strumentazione si trovi sempre allo stato ottimale per lo sviluppo del progetto. Questo riguarda ma non si limita a prendersi cura del proprio hardware, assicurarsi che i software siano sempre aggiornati all'ultima versione, *et cetera*.

F5. Quality Control

Ogni progresso eseguito nella realizzazione del progetto viene supervisionato e/o notificato agli altri membri, che in date prestabilite o per propria iniziativa eseguiranno un controllo di coerenza tra il lavoro fatto fino a quel punto e l'ultimo aggiornamento. Lo scopo di questa attività è garantire il rispetto di quanto accordato nel Piano di Progetto.

F6. Testing

Per appurare la correttezza delle funzionalità implementate, è necessario testarne il funzionamento tramite simulazioni di utilizzo reale.

Per quanto riguarda le attività del progetto, sono state delineate le seguenti:

A1. Definizione gruppo di lavoro e idea iniziale

La fase iniziale del progetto differisce dalle attività successive in quanto la produzione concreta è limitata e si concentra di più sull'impostazione generale delle fondamenta del progetto.

A2. Analisi sulla possibilità di realizzazione

Per effettuare scelte di progetto consapevoli, è necessario capire fino a che punto le nostre risorse possono spingerci a lavorare, e delimitare le idee sulla base della disponibilità economica, della conoscenza degli strumenti e del tempo a nostra disposizione.

A3. Definizione progettuale

Il risultato delle due attività precedenti ci permette di procedere a capire in che modo proseguire nello sviluppo del progetto.

Tale fase prevede le seguenti *task*:

A3-1. Definizione dei ruoli all'interno del progetto: al fine di distribuire in modo equo e bilanciato il carico di lavoro tra i membri del team

A3-2. Analisi dei requisiti: vengono effettuate delle scelte di sviluppo relative ai requisiti necessari alla creazione dell'applicazione.

A3-3. Analisi dei rischi: vengono analizzati i possibili rischi e fattori di ritardo nello sviluppo.

A3-4. Pianificazione delle attività: vengono analizzate e definite le attività, le funzioni e le *task* individuali.

A4. Progettazione dell'interfaccia grafica

Delineare un prototipo di interfaccia grafica fin dai primi stadi dello sviluppo può portare ad una migliore definizione della gestione delle fasi di sviluppo successive.

A5. Progettazione della prima *release*

Questa attività segna l'inizio del percorso di sviluppo applicativo vero e proprio. Anch'esso è costituito da diverse *task*:

A5-1. Progettazione del database: viene progettata una base di dati per l'offerta dei dati all'utente.

A5-2. Definizione di UI e procedure fondamentali: fase che prevede la stesura di codice per definire una base concreta dell'applicazione.

A5-3. Sviluppo logica applicativa: avviene qui la scrittura della logica applicativa dell'applicazione, con l'implementazione delle relative strutture dati e algoritmi.

A5-4. Interfacciamento UI - Logica applicativa: i risultati parziali dati da sviluppo UI e sviluppo logica applicativa convergono in questa fase per consolidare l'applicazione.

A5-5. Completamento e ampliamento dell'applicazione: vengono effettuati gli ultimi ritocchi e vengono aggiunte funzionalità all'applicazione.

A6. Progettazione della *release* definitiva

In modo analogo all'attività **A5**, e in particolare alla *task* **A5-5**, arrivati a questa fase sarà necessario fare uso delle ultime settimane per ultimare lo sviluppo dell'applicazione, sistemare eventuali *bug* e aggiungere le ultime funzionalità in vista della consegna finale.

A7. Revisione e consegna finale

L'ultimo passo riguarda una revisione finale complessiva di codice e documentazione, per assicurarci che, una volta consegnata, l'applicazione non presenti problemi noti e offra tutte le funzionalità promesse.

Le *milestones* del progetto sono definite dalle due consegne:

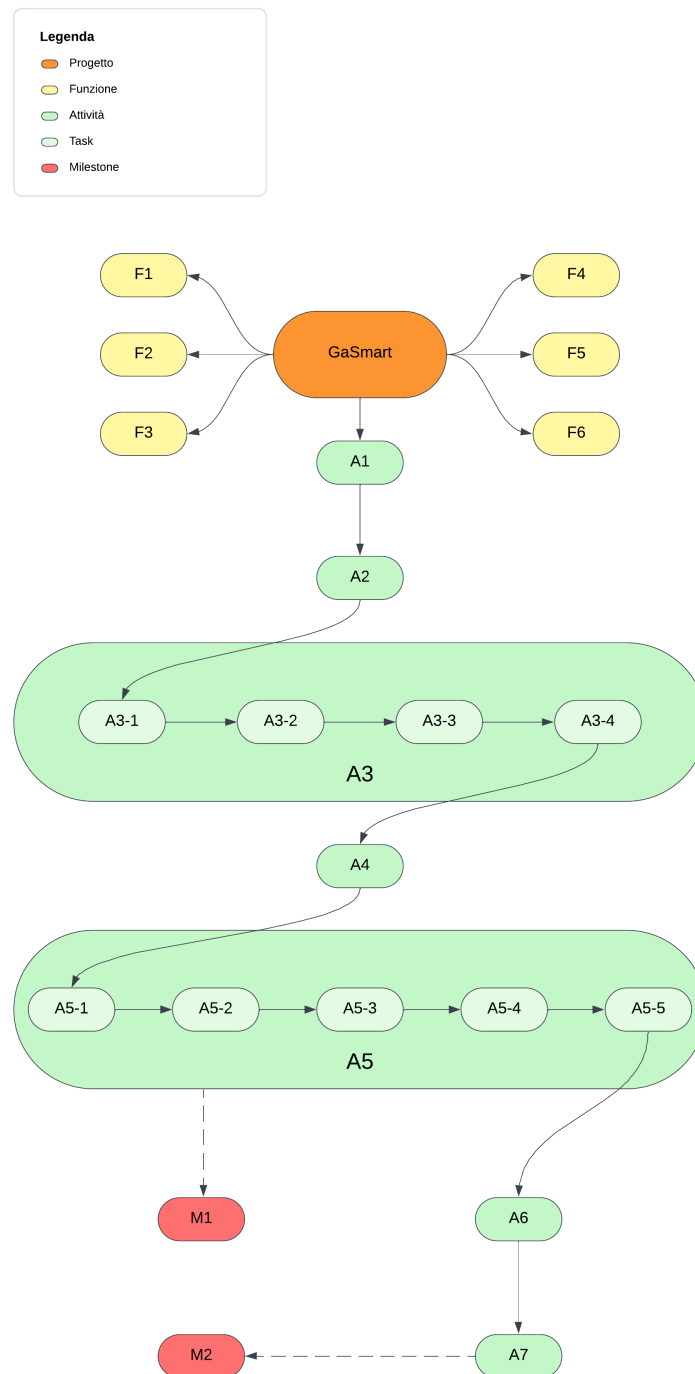
M1. Consegna della prima *release*

La versione 1.0 dell'applicazione dovrà aver implementato le principali funzionalità discusse e approvate dal team e proposte nella documentazione.

M2. Consegna finale

È previsto anche l'invio di una versione aggiornata dell'applicazione, che risolva eventuali *bug* presenti nella prima versione consegnata e che implementi, qualora ce ne fossero, nuove funzionalità.

Viene di seguito schematizzata l'analisi delle attività, funzioni e *task*.



5.2 Dipendenze

La natura variegata di attività e funzioni dovuta alla loro diversa finalità non permette di definire una rapporto di dipendenza lineare tra di esse: seppur sia ben chiaro delineare un punto d'inizio e un punto di fine del nostro progetto, non possiamo dire lo stesso delle attività intermedie, in quanto, *e.g.*, la stesura dei documenti è necessariamente concorrenziale alla stesura del codice nonché dell'attività di training finalizzata all'apprendimento del linguaggio.

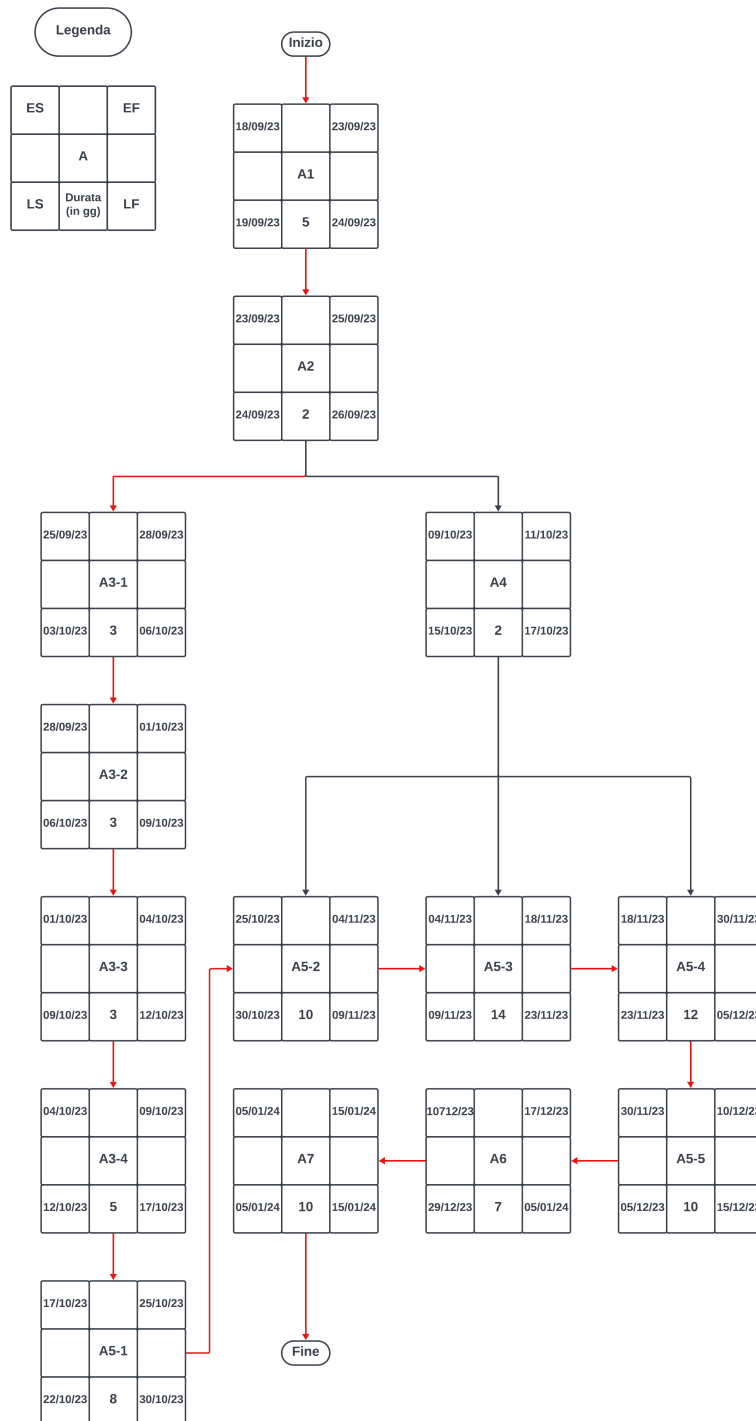
Al fine di delinare l'andamento temporale del progetto, viene proposto in diagramma di Pert che descrive i rapporti di dipendenza tra le diverse attività e *task*; in un diagramma di Pert, ad ogni attività/*task* vengono associati un numero di giorni necessari al suo compimento e una quadrupla di date, definite nel seguente modo:

- **ES - Earliest Start Time:** indica il minimo giorno di inizio dell'attività, a partire dal minimo tempo necessario per le attività che precedono;
- **EF - Earliest Finish Time:** dato ES e la durata dell'attività, indica il minimo giorno in cui l'attività può terminare;
- **LF - Latest Finish Time:** indica il giorno massimo in cui tale attività deve concludersi per non provocare ritardi nel progetto;
- **LS - Latest Start Time:** dato LF e la durata dell'attività, indica il giorno massimo entro il quale quell'attività deve iniziare per non provocare ritardi nel progetto.

Per costruire il diagramma si è fatto uso della seguente tabella temporanea per immagazzinare i dati.

| Attività/Task | Dipendenze | Giorno di inizio | Durata (in gg) | ES | EF | LS | LF |
|---------------|------------|------------------|----------------|------------|------------|------------|------------|
| A1 | - | 0 | 5 | 18/09/2023 | 23/09/2023 | 19/09/2023 | 24/09/2023 |
| A2 | A1 | 5 | 2 | 23/09/2023 | 25/09/2023 | 24/09/2023 | 26/09/2023 |
| A3 | A2 | 7 | 14 | 25/09/2023 | 09/10/2023 | 03/10/2023 | 17/10/2023 |
| A3-1 | A2 | 7 | 3 | 25/09/2023 | 28/09/2023 | 03/10/2023 | 06/10/2023 |
| A3-2 | A2 | 10 | 3 | 28/09/2023 | 01/10/2023 | 06/10/2023 | 09/10/2023 |
| A3-3 | A2 | 13 | 3 | 01/10/2023 | 04/10/2023 | 09/10/2023 | 12/10/2023 |
| A3-4 | A2 | 16 | 5 | 04/10/2023 | 09/10/2023 | 12/10/2023 | 17/10/2023 |
| A4 | A2 | 21 | 2 | 09/10/2023 | 11/10/2023 | 15/10/2023 | 17/10/2023 |
| A5 | A3 | 23 | 54 | 17/10/2023 | 10/12/2023 | 22/10/2023 | 15/12/2023 |
| A5-1 | A3 | 23 | 8 | 17/10/2023 | 25/10/2023 | 22/10/2023 | 30/10/2023 |
| A5-2 | A3, A4 | 31 | 10 | 25/10/2023 | 04/11/2023 | 30/10/2023 | 09/11/2023 |
| A5-3 | A4 | 41 | 14 | 04/11/2023 | 18/11/2023 | 09/11/2023 | 23/11/2023 |
| A5-4 | A3, A4 | 55 | 12 | 18/11/2023 | 30/11/2023 | 23/11/2023 | 05/12/2023 |
| A5-5 | A3 | 67 | 10 | 30/11/2023 | 10/12/2023 | 05/12/2023 | 15/12/2023 |
| A6 | A5 | 77 | 7 | 10/12/2023 | 17/12/2023 | 29/12/2023 | 05/01/2024 |
| A7 | A6 | 84 | 10 | 05/01/2024 | 15/01/2024 | 05/01/2024 | 15/01/2024 |

È evidenziato in rosso, nel diagramma di Pert, il cammino critico.



5.3 Risorse Necessarie

Vengono di seguito riportate le principali risorse necessarie allo sviluppo del progetto.

- **Risorse Fisiche**

In questa sezione sono raccolte le risorse di cui il team fa un uso concreto per la realizzazione del progetto. Dal momento che l'obiettivo finale del progetto è lo sviluppo di un software, le Risorse Fisiche possono essere viste come l'unione delle seguenti sottocategorie:

- **Risorse Hardware**

Con Risorse Hardware intendiamo le macchine calcolatrici dei singoli membri del team, ciascuna abbastanza performante per permettere l'esecuzione di Android Studio, oltre ad alcuni dispositivi mobili dotati del sistema Android che verranno utilizzati in fase di testing.

- **Risorse Software**

Con Risorse Software raggruppiamo la totalità dei software utilizzati per lo sviluppo del codice applicativo, della relativa documentazione e dei documenti *deliverable*, nonché dei servizi di messaggistica e di condivisione di codice utilizzati dal team per coordinare il lavoro di gruppo.

Maggiori informazioni sui singoli software utilizzati sono presenti nella sezione 1.4.1.

- **Risorse Umane**

Data la dimensione ristretta del progetto, possiamo racchiudere nell'insieme delle Risorse Umane i cinque membri del team di sviluppo e i futuri tester dell'applicazione.

- **Risorse Temporal**

Il tempo, in questo progetto, consiste nell'unica risorsa che richiede una gestione consapevole: se le risorse fisiche e umane sono certe, non possiamo dire lo stesso del tempo, in quanto ogni membro del team deve poter bilanciare la quantità di lavoro richiesta da questo progetto con lo studio e lo sviluppo di altri corsi e progetti. Una digressione più approfondita verrà eseguita nel paragrafo successivo.

5.4 Allocazione del Budget e delle Risorse

Data la natura del progetto, inteso come oggetto d'esame del corso di Ingegneria del Software, il team di sviluppo ha preso la decisione di considerare il fattore economico non determinante per lo sviluppo dell'applicazione: al momento dell'inizio del progetto, ogni membro era già in possesso delle Risorse Hardware necessarie, e la quasi totalità delle Risorse Software è infatti disponibile gratuitamente.

L'unica questione che potrebbe rendere rilevante il fattore economico riguarda l'utilizzo delle API di Google Maps: Google, infatti, mette a disposizione gratuitamente le sue API fintanto che le applicazioni che le implementano non superano la quota delle centomila richieste al mese; realisticamente, il team non si aspetta di superare tale soglia nel prossimo futuro, tantomeno entro il 15 gennaio 2024, data della consegna finale del progetto.

Tuttavia, al fine di realizzare questo progetto come se il team facesse parte di un'azienda di sviluppo software, possiamo compiere una digressione stimando un preventivo da fornire ad un possibile committente dell'applicazione. Ponendo come data di inizio del progetto il 18 settembre 2023, giorno in cui è stato creato il team, e come data di fine il 15 gennaio 2024, data di consegna della versione 1.1 del codice, contiamo tra le due date un totale di 17 settimane; supponendo un contributo da parte di ciascun membro di 12 ore di lavoro a settimana, e supponendo che lo stipendio medio per un programmatore in Italia sia di €16 all'ora, possiamo concludere che il solo costo per gli stipendi del team ammonta a circa €16.320.

5.5 Pianificazione

Sono di seguito elencate le *deadlines* per le consegne della documentazione.

1. Definizione dei gruppi di lavoro - 26/09/2023
2. Proposta iniziale - 03/10/2023
3. Piano di Progetto - 17/10/2023
4. Documento dei Requisiti - 31/10/2023
5. Piano di Testing - 14/11/2023
6. Documento di Progettazione - 28/11/2023
7. Versione 1.0 del codice sorgente - 15/12/2023
8. Versione 1.1 del codice sorgente e allineamento documentazione - 15/01/2024

Le funzioni del progetto sono costanti nel tempo: alcune iniziano prima di altre (*e.g.*, Project e Configuration Management cominciano prima di Testing e Quality Control), tuttavia ciascun membro del gruppo si occupa dell'adempimento costante delle funzioni per un periodo che va, indicativamente, dalla seconda settimana di ottobre fino al termine ultimo della consegna.

Per quanto concerne attività e *task*, il seguente diagramma di Gantt illustra graficamente la distribuzione del tempo da attribuire a ciascuna attività del programma.

