

PROGETTO INGEGNERIA DEL SOFTWARE  
A.A. 2023/24

D1: Piano di Progetto  
Versione 1.1

**MOMI**

15/01/2024

## Document Information

<b>Nome Progetto</b>	<b>My Food Tracker</b>
<b>Deliverable</b>	Piano di Progetto
<b>Consegna</b>	17/10/2023
<b>Team Leader</b>	Alessandro Campagnaro - <a href="mailto:863419@stud.unive.it">863419@stud.unive.it</a>
<b>Componenti</b>	Elia Corò - <a href="mailto:892507@stud.unive.it">892507@stud.unive.it</a> Lorenzo Vaccher - <a href="mailto:891921@stud.unive.it">891921@stud.unive.it</a> Davide Vecchiato - <a href="mailto:890998@stud.unive.it">890998@stud.unive.it</a>

## Document History

Version	Issue Date	Stage	Description	Contributors
1.0	17/10/2023	Draft	Creazione Piano di Progetto	Alessandro Campagnaro Elia Corò Davide Vecchiato Lorenzo Vaccher
1.1	15/01/2024	Final	Allineamento documentazione per consegna	Alessandro Campagnaro Elia Corò Davide Vecchiato Lorenzo Vaccher

# Indice

<b>1. Introduzione</b>	<b>4</b>
1.1. Overview del Progetto	4
1.2. Deliverables del Progetto	4
1.3. Evoluzione	5
1.4. Materiale di Riferimento	5
1.5. Definizioni ed Abbreviazioni	6
<b>2. Organizzazione del Progetto</b>	<b>7</b>
2.1. Modello del Processo	7
2.2. Struttura Organizzativa	7
2.3. Interfacce Organizzative	8
2.4. Responsabilità di Progetto	8
<b>3. Processi Gestionali</b>	<b>9</b>
3.1. Obiettivi e Priorità	9
3.2. Assunzioni, Dipendenze , Vincoli	9
3.3. Gestione Rischi	10
3.4. Monitoraggio e Controllo	13
3.5. Pianificazione dello Staff	13
<b>4. Processi Tecnici</b>	<b>14</b>
4.1. Metodi, Strumenti, Tecniche	14
4.2. Documentazione Software	14
4.3. Funzionalità di Supporto al Progetto	14
<b>5. Pianificazione Lavoro, Risorse Umane e Budget</b>	<b>15</b>
5.1. WBS - Work Breakdown Structure	15
5.2. Dipendenze	16
5.3. Risorse Necessarie	17
5.4. Allocazione Budget e Risorse	17
5.5. Pianificazione	18

# Introduzione

## 1.1 Overview del Progetto

Il progetto consiste nella creazione di un'applicazione Android in cui viene messa a disposizione una lista di tutti i prodotti alimentari (confezionati) che fornisce per ognuno di essi informazioni come valori nutrizionali, allergeni, smaltimento ed altre informazioni utili.

L'applicazione permetterà di:

- Registrazione e Login all'applicazione tramite email/password o con account Google.
- Scansione codice a barre per consultazione informazioni prodotto.
- Creazione di una dispensa e registrazione degli alimenti in essa presenti, inserimento data di scadenza e sistema di notifica all'avvicinarsi di quest'ultima.
- Creazione lista di prodotti da acquistare con aggiunta prodotto e quantità tramite scansione codice a barre sopra citato, ricerca manuale o a partire dai prodotti già presenti nella dispensa dell'utente nell'applicazione.
- Consultazione informazioni di ogni prodotto ricercato o scansionato.
- Possibilità di inserimento quantità consumate per ogni prodotto ed aggiornamento quantità presenti nella dispensa.

## 1.2 Deliverables del Progetto

Nome	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 Codice Sorgente	15/12/2023	Moodle del Corso
Versione 1.1 Codice Sorgente ed allineamento documentazione	15/01/2024	Moodle del Corso

### 1.3 Evoluzione

Essendo ancora nelle fasi di definizione iniziale del progetto, non siamo ancora in grado di identificare dei possibili cambiamenti all'applicazione e alle sue funzionalità, ma useremo le informazioni ottenute tramite un sondaggio creato nella fase di Analisi di Mercato per migliorare le funzionalità già elencate o implementarne di nuove.

### 1.4 Materiale di Riferimento

- [Slides del corso di Ingegneria del Software](#)
- [Documenti e materiale del Team Breaking Bad riguardanti il progetto di Ingegneria del Software oggetto d'esame nell'A.A. 2022/2023](#)
- [Documentazione Firebase](#)
  - [Autenticazione](#)
  - [Firestore](#)
  - [Crashlytics](#)
  - [Monitoraggio delle prestazioni](#)
  - [Analytics](#)
  - [Remote config](#)
  - [Cloud Messaging](#)
- [Documentazione Flutter](#)
- [Librerie per Dart e Flutter](#)
- [Template](#)
- [Documentazione Android Studio per emulatore](#)
- [StackOverflow: forum Q&A per soluzioni a problemi comuni riscontrati da altre persone.](#)

## 1.5 Definizioni ed Abbreviazioni

Lista di concetti e descrizioni ricorrenti che definiremo in questa sezione del documento e per brevità riassumeremo con:

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

**Android**: Android è un sistema operativo per dispositivi mobili sviluppato da Google, progettato principalmente per sistemi embedded quali smartphone e tablet, con interfacce utente specializzate per televisori, automobili, orologi da polso, occhiali, e altri.

**Google Drive**: Google Drive è un servizio web, in ambiente cloud computing, di memorizzazione e sincronizzazione online.

**Documenti Google**: Google Docs è un elaboratore di testi online incluso come parte della suite gratuita di editor di documenti Google basata sul Web offerta da Google.

**Kanban**: In ambito di sviluppo software si fa riferimento a Kanban come una metodologia di sviluppo software, associata al processo Agile. Questo metodo punta a migliorare il flusso di lavoro suddividendo i compiti in fasi più piccole da elaborare una alla volta in sequenza.

**Whatsapp**: Applicazione di messaggistica istantanea per dispositivi mobili multiplatforma che, attraverso la connessione a Internet, consente lo scambio tra uno o più utenti di messaggi di testo e file multimediali.

**Visual Studio Code**: Visual Studio Code (o più semplicemente VS Code) è un editor di codice sorgente sviluppato da Microsoft per Windows, Linux e macOS.

**ML**: Il Machine Learning (ML) è un sottoinsieme dell'intelligenza artificiale che si occupa di creare sistemi che apprendono o migliorano le performance in base ai dati che utilizzano.

**IDE**: un Integrated Development Environment, ovvero un ambiente di sviluppo integrato, è un software che in fase di programmazione aiuta gli sviluppatori con strumenti integrati per facilitare lo sviluppo ed il debugging del codice.

# ORGANIZZAZIONE DEL PROGETTO

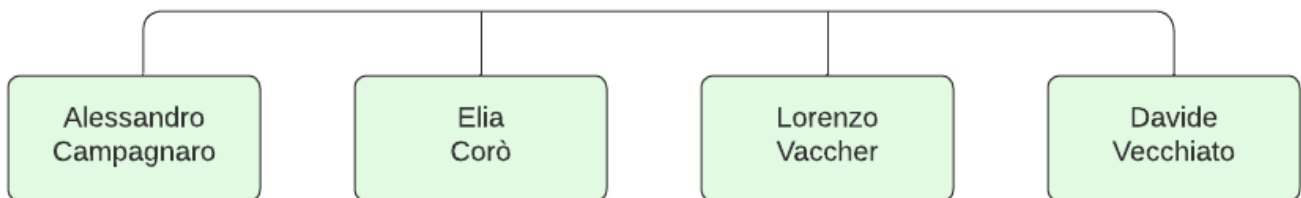
## 2.1 Modello del Processo

Il modello di processo adottato per questo progetto è Agile, il quale ha permesso di gestire in modo più dinamico le varie fasi di sviluppo e progettazione, dando importanza ad un codice sicuro e funzionante piuttosto che ad un codice che necessiti di molta documentazione e che allo stesso tempo permetterà di soddisfare al meglio le necessità degli utenti.

## 2.2 Struttura Organizzativa

Il gruppo ha deciso di adottare per questo progetto, una struttura organizzativa per funzioni di tipo democratico decentralizzato nella quale non è presente un'unica figura a cui spetta tutto il controllo proprio per poter ripartire in maniera equa il potere decisionale e anche grazie alla dimensione ridotta del Team di Progetto per poter avere una comunicazione più efficace di tipo peer-to-peer.

Il team sarà ripartito internamente in base alle varie funzioni da gestire ma ogni membro parteciperà attivamente e in egual misura alle decisioni.



## 2.3 Interfacce Organizzative

Durante tutto il periodo di sviluppo dell'App ci sarà il bisogno di interagire con entità esterne al team. In caso di necessità il gruppo dovrà interagire con il Professor Agostino Cortesi per quanto riguarda la parte organizzativa e di progettazione oppure con il Professor Alvise Spanò per chiarimenti riguardo alla parte tecnica e di programmazione.

Ci sarà inoltre una fase di Analisi di Mercato e una fase di Testing che richiederanno di interfacciarsi con possibili e/o futuri utenti per una miglior definizione dei requisiti e delle funzionalità e per avere un riscontro su eventuali problemi sorti durante le prime fasi di utilizzo dell'App.

## 2.4 Responsabilità di Progetto

Si è scelto di dividere il gruppo in due team che si occuperanno ciascuno di una task diversa ma dai quali è prevista una mutua collaborazione anche per il decision-making e la risoluzione di eventuali problemi.

All'interno dei due team, i membri saranno comunque alla pari senza necessità di individuare un supervisore.

Il gruppo è stato diviso nella seguente maniera:

- Alessandro Campagnaro: redazione ed aggiornamento documentazione
- Davide Vecchiato: aggiornamento documentazione + sviluppo codice sistema notifiche
- Elia Corò: sviluppo codice, gestione DevOps
- Lorenzo Vaccher: sviluppo codice, gestione DevOps



# PROCESSI GESTIONALI

## 3.1 Obiettivi e Priorità

Per la buona riuscita del progetto, il gruppo si impone di:

- rispettare le scadenze stabilite
- riuscire a collaborare mantenendo una linea di lavoro chiara ed organizzata.
- produrre un'App qualitativamente buona e che permetta una buona usabilità.

## 3.2 Assunzioni, Dipendenze e Vincoli

### Assunzioni

Si assume che tutti i membri del gruppo siano al corrente delle fasi di lavoro e delle deadlines da rispettare. Inoltre assumiamo che i membri del gruppo seguano attivamente le lezioni del Professor Cortesi e del Professor Spanò per potere avere una buona comprensione di obiettivi e metodo di lavoro.

Ci aspetta infine, che ogni membro del gruppo dedichi il tempo e l'impegno necessario al completamento dei task ad egli assegnati per permettere la buona riuscita del progetto.

### Dipendenze

L'App che il Team andrà a sviluppare si appoggia ad un database creato tramite Firebase su cui sarà necessario effettuare operazioni di manutenzione periodiche. Questo servizio ci permette, di ridurre i tempi previsti per lo sviluppo dell'applicazione.

### Vincoli

Sono stati definiti dei vincoli temporali e strutturali, ovvero le deadlines per la consegna dei deliverable e la loro struttura, fissate dal Professor Cortesi. Si identifica inoltre come vincolo la creazione di App sicura con un buon livello di usabilità.

### 3.3 Gestioni dei Rischi

Individuiamo come principali rischi in cui potremmo incorrere durante lo sviluppo del progetto:

1. Guasti Software: il gruppo ritiene che guasti di questo genere siano risolvibili in poco tempo e non vadano a danneggiare gravemente il progetto, richiedendo in linea di massima una reinstallazione del software.
2. Guasti Hardware: in caso si dovessero verificare guasti hardware ad uno dei computer utilizzati dai membri del gruppo, saremo in grado di utilizzarne un altro, non incorrendo quindi in perdite di entità grave, presupponendo come perdita massimo il lavoro eseguito su quella macchina dall'ultimo salvataggio su Github fino al momento di guasto.
3. Problemi di Salute: il caso più grave in cui tutti i membri del gruppo siano affetti da problemi di salute potrebbe causare rallentamenti significativi. Tuttavia si prevede, in casi meno gravi, il mutuo aiuto in modo da sopperire ai rallentamenti e alle necessità dei membri indisposti.
4. Perdita involontaria progressi: consideriamo che questo problema andrebbe ad inficiare in maniera non superficiale le tempistiche di sviluppo del progetto, tuttavia il gruppo ritiene di essere preparato all'evenienza tenendo aggiornata la repository Github e la cartella di lavoro di Google Drive relative al progetto grazie alle quali sarà possibile recuperare codice e documenti andati persi o cancellati.
5. Bugs a livello Codice/App: sarà quasi inevitabile non incontrare problemi del genere, proprio per questo motivo il gruppo si munisce di strumenti come StackOverflow per trovare delle soluzioni online e prevede delle fasi di testing periodiche per assicurarsi che il lavoro svolto sia privo di bugs ed in caso contrario correggerli tempestivamente.
6. Difficoltà Linguaggio: verosimilmente ci saranno problemi riguardanti peculiarità del linguaggio di programmazione utilizzato per sviluppare l'App che causeranno rallentamenti. Al fine di risolvere questi problemi il gruppo si impegna al mutuo aiuto e a colmare le lacune attraverso l'utilizzo e la comprensione della documentazione.

### Tabella dei Rischi

ID	RISCHI	CATEGORIA	PROBABILITÀ	IMPATTO	AZIONE
1	Guasti SW	Sviluppo	2	2-4	Disinstallazione e Reinstallazione Software
2	Guasti HW	Materiale	1	2	Riparare la Macchina e/o trovare macchina sostitutiva
3	Problemi di Salute	Personale	3	1-3 (per problemi di lieve entità)	Suddivisione carico di lavoro della persona impossibilitata.
4	Perdita Progressi	Sviluppo	2	4-5	Recuperare ultima versione salvata dai backup online.
5	Bugs	Sviluppo	5	1-5	Trovare cause e modificare codice adeguatamente.
6	Difficoltà Linguaggio	Sviluppo	2	2	Studio documentazione

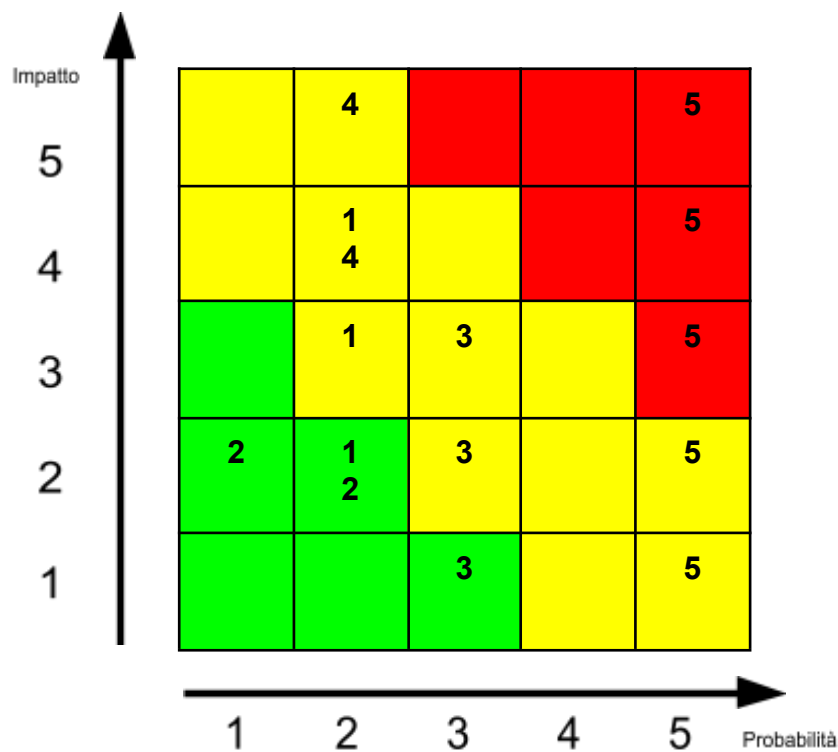
#### Legenda:

- 1: Probabilità molto bassa
- 2: Probabilità bassa
- 3: Probabilità media
- 4: Probabilità alta
- 5: Probabilità molto alta

## Prevenzione e Monitoraggio dei Rischi

ID	RISCHIO	CATEGORIA	PREVENZIONE	MONITORAGGIO
1	Guasti SW	Sviluppo	Installare ed eseguire in maniera sicura i software utilizzati sulla propria macchina	Tenere sotto controllo eventuali anomalie
2	Guasti HW	Materiale	Utilizzare e riporre la propria macchina in luoghi al riparo da luoghi o situazioni dannose	Controllo integrità hardware
3	Problemi di Salute	Personale	Evitare situazioni di stress eccessivo per non abbassare difese immunitarie ed evitare situazioni di rischio associato	Prestare attenzione all'insorgere di eventuali sintomi
4	Perdita Progressi	Sviluppo	salvataggio codice e documenti in backup online	controllare ed accertarsi dello stato della sincronizzazione
5	Bugs	Sviluppo	scrivere il codice in maniera chiara e consapevole	eseguire controlli periodici su scrittura codice (attraverso ad esempio strumenti integrati IDE)
6	Difficoltà Linguaggio	Sviluppo	apprendimento specifico linguaggio e consultazione documentazione	controllo periodico codice

## Modello dei Rischi



### 3.4 Meccanismi di Monitoraggio e Controllo

Ci si aspetta che il monitoraggio e il controllo sia fatto da ciascun membro a cui è stato assegnato un determinato task da eseguire ma allo stesso tempo ci saranno controlli periodici dove tutti i membri del gruppo sono tenuti a controllare l'operato degli altri ed informarli in caso di problemi. In necessità di notifica tempestiva e nell'impossibilità di incontrarsi di persona verrà utilizzata Whatsapp come piattaforma di comunicazione tra i membri del gruppo.

Per problemi meno urgenti verranno invece utilizzate le modalità di Commento e Suggerimento dei Documenti di Google e la tabella Kanban di Github.

### 3.5 Pianificazione dello Staff

La realizzazione di questo progetto richiede la preparazione dei membri del gruppo su Dart, Flutter, la conoscenza e la gestione di basi di dati e la familiarità con Visual Studio Code con aggiunta del simulatore di dispositivi di Android Studio. Inoltre, è necessaria la conoscenza di Github.

## PROCESSI TECNICI

### 4.1 Metodi, Strumenti e Tecniche

L'ambiente di sviluppo che verrà utilizzato per lo sviluppo del progetto è Visual Studio Code che permette di sviluppare applicazioni Android come richiesto dalle specifiche del progetto. Viene inoltre utilizzato l'emulatore di dispositivi mobili di Android Studio per permettere ai membri del gruppo non in possesso di dispositivi Android di testare l'applicazione. Consapevoli delle differenze tra dispositivi fisici e virtuali, l'applicazione verrà anche testata su dispositivi Android fisici prima del rilascio in modo da poter rilevare eventuali problematiche non insorte tramite il testing su emulatore.

Il linguaggio di programmazione utilizzato per lo sviluppo è Dart attraverso l'utilizzo del framework Flutter.

Lo sviluppo è stato suddiviso in tasks per alleviarne il peso e poter fare dei controlli periodici sul progresso raggiunto.

### 4.2 Documentazione del Software

Il Progetto sarà dotato di documentazione che si suddivide in:

- Piano di Progetto
- Documento dei Requisiti
- Piano di Testing
- Documento di Progettazione

la cui stesura finale verrà consegnata in data 15/01/2024 per necessità di aggiornamento e modifica degli stessi durante lo sviluppo.

### 4.3 Funzionalità di Supporto al Progetto

Identifichiamo come funzionalità di supporto:

- l'aggiornamento costante della documentazione
- la rilevazione e risoluzione di bug o anomalie a livello applicativo tramite debugging e testing

Identifichiamo come strumenti di supporto allo sviluppo:

- Github
- Visual Studio Code
- Whatsapp
- Documenti Google

# PIANIFICAZIONE LAVORO, RISORSE UMANE E BUDGET

## 5.1 WBS - Work Breakdown Structure

Il team definisce come principali funzioni:

- **Project Management:**  
occuperà tutta la durata dello sviluppo del progetto e servirà a tenere traccia di problematiche, modifiche e progressi relativi al progetto.
- **Training:**  
studio della documentazione ed apprendimento dei linguaggi e dell'utilizzo degli strumenti utili allo sviluppo. Quindi apprendimento Dart ed utilizzo di Visual Studio Code, simulatore Android Studio, Github, database ed API Firebase.
- **Documenti:**  
stesura dei documenti necessari e utili allo sviluppo. Verranno redatti di volta in volta in base alle scadenze prefissate e utilizzando il formato strutturale indicato dal Professor Cortesi. I documenti verranno consegnati rispettando le scadenze prestabilite ma, se necessarie delle modifiche verranno aggiornati al bisogno. I documenti saranno aggiornati con data ultima la consegna finale del progetto 15/01/2024.

I documenti che verranno consegnati sono stati specificati alla sezione 4.2 .

- **Quality Control e Testing:**  
Controlli periodici su bugs ed anomalie e fase di Testing per assicurare il corretto funzionamento dell'App.

Il team definisce come principali attività:

### A. Ideazione Progetto

1. Condivisione e discussione idee alla base della realizzazione del progetto.
2. Identificazione e definizione proposta di progetto.
3. Definizione struttura organizzativa team/progetto.
4. Definizione ruoli e compiti all'interno del Team.
5. Identificazione funzionalità e struttura dell'applicazione.

### B. Analisi di Fattibilità

1. Definizione Piano di Progetto
2. Identificazione dei vincoli, dei requisiti e degli obiettivi del progetto.

### **C. Progettazione Applicazione:**

1. Configurazione Database con applicazione
2. Ideazione e sviluppo UI e struttura front-end dell'applicazione.
3. Creazione sistema di notifica.
4. Configurazione API con applicazione.
5. Prototipo Applicazione (Versione 1.0 - MILESTONE)

### **D. Attività di Ultimazione Progetto**

1. Ultimazione e consegna Progetto. Versione finale Applicazione.  
(Versione 1.1 - MILESTONE)

Il team definisce come proprie Milestones:

- **Versione 1.0**

Versione completa del progetto ad App sviluppata e funzionalità di base implementate e possibilità di rilascio agli utenti per feedback e miglioramento in vista della consegna finale.

- **Versione Finale 1.1**

Progetto ultimato con implementazione feedback utenti tester, risoluzione bug e aggiornamento finale documentazione di progetto.

## **5.2 Dipendenze**

Sono state riconosciute come dipendenze tutte le attività di lavoro non parallelizzabili che necessitano delle attività precedenti per essere messe in atto. Il team ha comunque adottato il modello Agile per poter lavorare contemporaneamente a sub-tasks indipendenti all'interno di ogni macro-attività dipendente da altre.

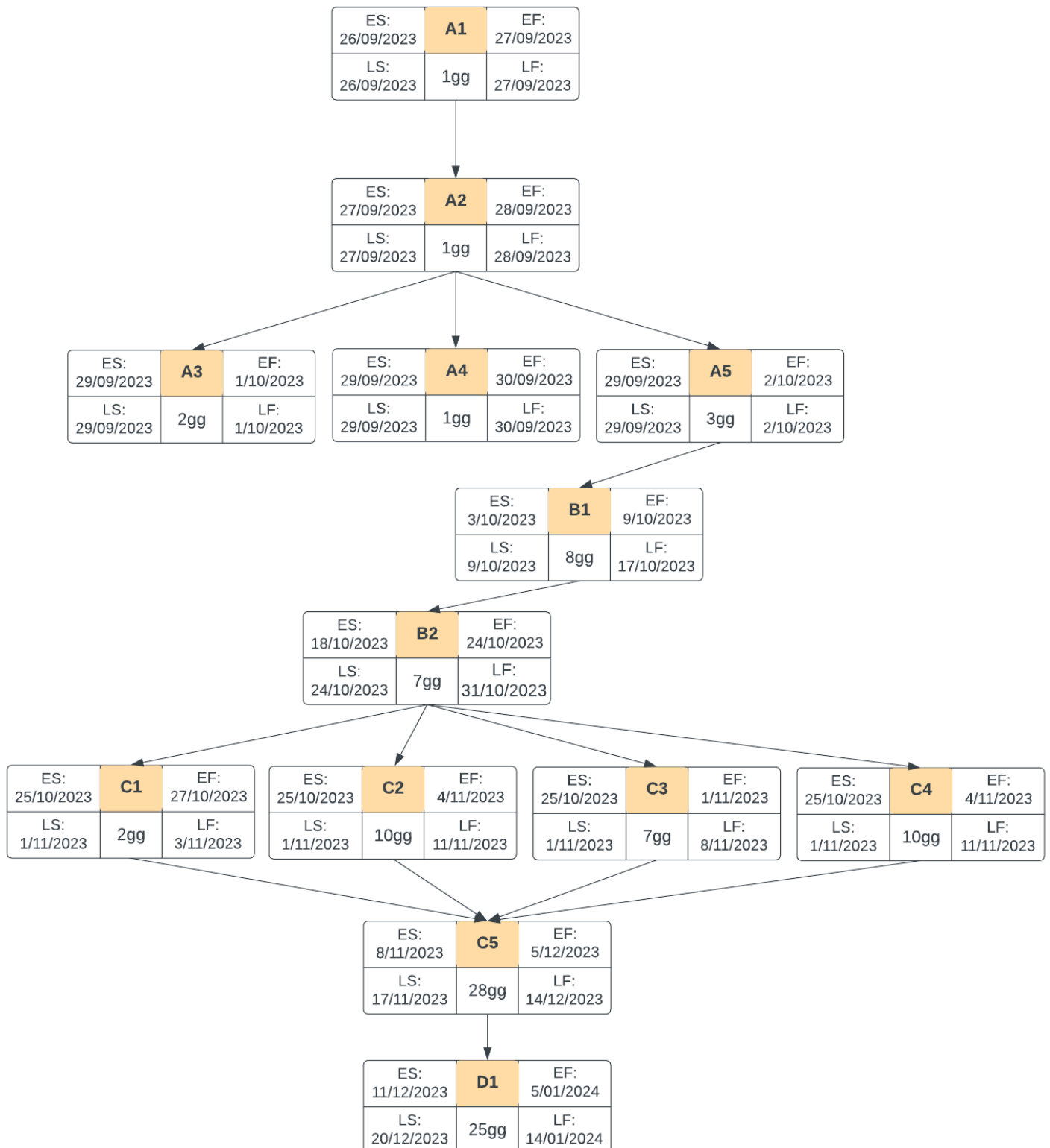


Viene riportata di seguito una tabella per illustrare le dipendenze ovvero quelle attività vincolate l'una dall'altra e quindi non parallelizzabili.

Verranno riportate nella tabella Earliest Start Time (ES) ovvero la prima data disponibile all'inizio di una determinata attività, l'Earliest Finish Time (EF) ovvero la prima data disponibile per la conclusione dell'attività tenendo conto della durata dell'attività e della data di inizio ed anche Latest Start Time (LS) e Latest Finish Time (LF), date ultime per l'inizio e la fine dell'attività tali che l'attività non rallenti e metta in stallo le altre attività e conseguentemente l'avanzamento del progetto:

Attività	Dipendenza	Durata	ES	EF	LS	LF
A1	/	1gg	26/09/2023	27/09/2023	26/09/2023	27/09/2023
A2	A1	1gg	27/09/2023	28/09/2023	27/09/2023	28/09/2023
A3	A2	2gg	29/09/2023	1/10/2023	29/09/2023	1/10/2023
A4	A2	1gg	29/09/2023	30/09/2023	29/09/2023	29/09/2023
A5	A2	3gg	29/09/2023	2/10/2023	29/09/2023	2/10/2023
B1	A5	8gg	3/10/2023	11/10/2023	9/10/2023	17/10/2023
B2	B1	7gg	18/10/2023	24/10/2023	24/10/2023	31/10/2023
C1	B2	2gg	25/10/2023	27/10/2023	1/11/2023	3/11/2023
C2	B2	10gg	25/10/2023	4/11/2023	1/11/2023	11/11/2023
C3	B2	7gg	25/10/2023	1/11/2023	1/11/2023	8/11/2023
C4	B2	10gg	25/10/2023	4/11/2023	1/11/2023	11/11/2023
C5	C1-C2-C3-C4	28 gg	8/11/2023	5/12/2023	17/11/2023	14/12/2023
D1	C5	25gg	11/12/2023	05/01/2024	20/12/2023	14/01/2024

## Diagramma di Pert



## 5.3 Risorse Necessarie

### Risorse Hardware

Computer che abbiano la capacità e siano abbastanza performanti da poter utilizzare il simulatore di Android Studio e dispositivi Android fisici su cui eseguire parte della fase di testing.

### Risorse Software

Android Studio per utilizzare il simulatore di dispositivi mobili. Documenti Google per la stesura della documentazione relativa al progetto. Whatsapp per la comunicazione tra team members. Tutte le altre risorse elencate nella sezione 4.3.

### Risorse Umane

I team members.

### Risorse Temporal

Ore lavorative necessarie allo sviluppo del progetto.

## 5.4 Allocazione Budget e Risorse

Per la realizzazione del progetto il gruppo ha deciso di acquistare un template in modo da permettere di concentrarsi maggiormente sul progetto e la sua realizzazione e diminuire i tempi e le risorse impiegate per la scrittura di codice. L'utilizzo del template implica un costo una tantum di 18.02 €.

Attualmente, lo sviluppo delle funzionalità dell'App ci siamo affidati interamente agli strumenti interni di Firebase.

Per il momento, stiamo utilizzando Firebase - Spark Plan che non prevede costi iniziali di abbonamento ma consente un massimo di 50mila utenti attivi al mese, soglia oltre al quale si aggiungerebbe un costo di 0.0052 € ogni utente attivo al mese. Il database utilizzato per l'app è gestito tramite Cloud Firestore che consente di salvare dati fino ad un massimo di 1 GiB e si prevede un costo aggiuntivo di 0.17 € ogni GiB di archiviazione utilizzato al mese oltre la soglia.

Firestore impone infine un limite di 50mila letture e 20mila scritture documenti al giorno, oltre le quali verranno addebitati 0.052 € ogni 100.000 letture e 0.17€ ogni 100 scritture.

Per quanto riguarda invece i costi delle risorse, la durata complessiva dello sviluppo del progetto sarà di circa 17 settimane con settimana lavorativa di 5 giorni. Si stima un impegno di circa 2 ore al giorno da parte di tutti i team members per un ammontare di circa 680 ore complessive. Tenendo conto che lo stipendio medio di uno sviluppatore software è circa 16 €/h, stimiamo un preventivo di 10.900 €.

## 5.5 Pianificazione

### Diagramma di Gantt

