



Università  
Ca' Foscari  
Venezia

# INGEGNERIA DEL SOFTWARE

## PIANO DI PROGETTO

BeeSafe

Buffer Overflow

SERANI Hernest 877028

BALDASSO Enrico 874885

CAPPON Stefano 876895

# Indice

1. Introduzione
  - 1.1. [Overview del progetto](#)
  - 1.2. [Deliverables del progetto](#)
  - 1.3. [Evoluzione del progetto](#)
  - 1.4. [Materiale di riferimento](#)
  - 1.5. [Definizioni ed abbreviazioni](#)
  
2. Organizzazione del Progetto
  - 2.1. [Modello del processo](#)
  - 2.2. [Struttura organizzativa](#)
  - 2.3. [Interfacce organizzative](#)
  - 2.4. [Responsabilità di progetto](#)
  
3. Descrizione dei processi gestionali
  - 3.1. [Obiettivi e priorità](#)
  - 3.2. [Assunzioni, dipendenze e vincoli](#)
  - 3.3. [Gestione dei rischi](#)
  - 3.4. [Pianificazione dello staff](#)
  
4. Descrizione dei processi tecnici
  - 4.1. [Metodi, strumenti e tecniche](#)
  - 4.2. [Documentazione del software](#)
  - 4.3. [Funzionalità di supporto al progetto](#)
  
5. Pianificazione del lavoro, delle risorse umane e del budget
  - 5.1. [WBS - Work Breakdown Structure](#)
  - 5.2. [Dipendenze](#)
  - 5.3. [Risorse necessarie](#)
  - 5.4. [Allocazione del budget e delle risorse](#)
  - 5.5. [Pianificazione](#)

# 1. Introduzione

## 1.1 - Overview del progetto

**BeeSafe** - Il progetto del corso è stato scelto tra i progetti proposti dal tutor e consiste nel creare un'applicazione Android per gestire gli affollamenti e le affluenze in real-time, misurando la quantità di persone, avvisando quando si creano assembramenti.

### **Il funzionamento dell'app sarà così diviso:**

1. Un dispositivo utilizzando il bluetooth e scansiona altri dispositivi nelle vicinanze usando l'indirizzo MAC e i Beacon "lanciati" dagli altri dispositivi.
2. Se il numero di rilevamenti è maggiore di un certo numero, variabile alla fase di sviluppo (dobbiamo fare delle ricerche per approssimare il numero di dispositivi in un posto affollato) allora l'applicazione manda nel database sul server le coordinate GPS per segnalare che quel posto è affollato.
3. Siccome il numero di persone in un posto può cambiare continuamente, dopo un certo tempo senza nuove segnalazioni in quell'area, togliamo le coordinate GPS dal database perché il luogo potrebbe non essere più affollato.
4. Ogni utente nell'applicazione vedrà una mappa dinamica con degli heat map colorati, che indicano la densità stimata di certe coordinate. L'utente può anche mettere dei posti preferiti (come il suo bar preferito, il posto di lavoro, la fermata dell'autobus sotto casa etc). Un'altra feature pianificata è la predizione se un posto sarà affollato in un certo orario, basato sui dati che salveremo nel nostro database.
5. Se un utente si avvicina ad un posto affollato gli viene inviata una notifica.
6. L'algoritmo di valutazione dei dispositivi avrà altri parametri, compreso un sistema di filtro dei mac address rivelati (es. se io ho 3 dispositivi non voglio che l'algoritmo mi conti i miei dispositivi come dispositivi di altre persone).

## 1.2 - Deliverables del progetto

- Piano di Progetto (18/10/2020)
- Documento di analisi e specifica (27/10/2020)
- Piano di testing (14/11/2020)
- Documento di Progettazione (09/12/2020)
- Codice e messa in linea del sistema (15/01/2021)

## 1.3. - Evoluzione del progetto

Il progetto verrà suddiviso in diverse attività le quali verranno svolte in parallelo con gli altri membri del gruppo. Le principali attività sono:

- Pianificazione del progetto.
- Sviluppo e progettazione del database nel server (la parte backend)
- Sviluppo di un piccolo API tra l'applicazione e il server.
- Realizzazione dell'applicazione.
  - Parte front-end
    - Usare Material Design per la realizzazione di una pagina per modificare i luoghi preferiti e impostazioni generali.
    - Realizzazione di una mappa dinamica e le sue componenti.
  - Parte back-end
    - Sviluppo di un algoritmo per fare tracing di dispositivi che usano Bluetooth nelle vicinanze.
    - Comunicazione con il server per caricare e scaricare i dati della nostra applicazione usando l'API.
    - Collegamento tra i dati e la mappa dinamica.
- Test e debug dell'applicazione

## 1.4. - Materiale di riferimento

- Slides del corso Ingegneria del Software
- Piani di progetto anni precedenti
- Google
  - [API di Android Studio](#)
  - [Google Maps SDK](#)
  - [Material Design per Android](#)
  - [Firebase](#)

## 1.5. - Definizioni ed abbreviazioni

**Android** - Sistema operativo per dispositivi mobili sviluppato da Google.

**API** - Application Programming Interface - Insieme di protocolli e procedure per ricevere o trasmettere un determinato servizio.

**App** - Si intende l'applicazione ideata e descritta in questo documento.

**Beacon** - Pacchetti che mandano i dispositivi che usano la tecnologia Bluetooth per notificare agli altri dispositivi la loro presenza.

**Bluetooth** - Standard tecnico-industriale di trasmissione dati per reti personali senza fili.

**Bug** - Errore nella scrittura di un programma software.

**DB** - Database, ovvero insieme di informazioni registrate in una memoria di massa, gestite da un apposito programma.

**Discor** - Discord è un'applicazione VoIP utilizzata per effettuare chiamate/meet di gruppo.

**Drive** - Piattaforma di Google utilizza per la condivisione di documenti.

**Package** - Meccanismo per organizzare il codice in gruppi logici.

**Play Store** - Piattaforma digitale di distribuzione di applicazioni per Android gestita da Google.

**SDK** - Insieme di strumenti per lo sviluppo di un software

**Telegram** - Telegram è un servizio di messaggistica istantanea.

## **2. Organizzazione del progetto**

### 2.1 - Modello del processo

Per questo progetto abbiamo deciso di applicare il modello a cascata.

Le fasi del progetto saranno le seguenti:

- Analisi e definizione dei requisiti
- Progettazione del sistema e del software
- Implementazione e test delle singole unità
- Integrazione e test del sistema
- Installazione e mantenimento

## 2.2. - Struttura organizzativa

Il team è composto da 3 persone, le specifiche e le funzionalità verranno sempre discusse tutti insieme. Sostanzialmente tutti hanno lo stesso potere decisionale. Ogni idea di ciascun membro verrà presa in discussione seriamente, valutata da tutti, e alla fine dare la decisione usando il fondamento della democrazia, il voto. Successivamente ci divideremo il lavoro per la fase di debugging dell'applicazione. Non avendo delle conoscenze in macroaree specifiche abbiamo deciso di dividerci in ruoli come segue:

- **Hernest SERANI**

- Back-End
  - Algoritmi di tracing tramite Bluetooth
  - Gestione sensori GPS e Bluetooth
  - L'uso delle API di Maps
  - Programmazione dell'Applicazione
- Front-End
  - UI e UX Designer dell'Applicazione
  - Visualizzazione di Mappa interattiva
- Generale
  - Gestore, editore e designer dei deliverables

- **Enrico BALDASSO**

- Back-End
  - Programmazione dell'Applicazione
  - Gestore delle API dentro applicazione e per la comunicazione con il server
  - Algoritmi per dati statistici e predizione
  - L'uso delle API di Maps
- Generale
  - Gestione del progetto usando GitHub
  - Editore dei deliverables

- **Stefano Cappon**

- Back-End
  - Programmazione dell'Applicazione
  - Gestione del server
  - Gestione del DBMS
  - Comunicazione con il server tramite API
- Front-End
  - Interfacce Grafiche dell'Applicazione
  - Gestione delle notifiche dell'Applicazione
- Generale
  - Editore dei deliverables

## 2.3. - Interfacce organizzative

Le riunioni del gruppo verranno fatte tramite canali tecnologici. Useremo l'applicazione Telegram per la messaggistica istantanea e l'applicazione Discord per effettuare dei meet periodici per lavorare in contemporanea o aggiornarsi sulle modifiche effettuate. Se sarà possibile ci si organizzerà per incontrarsi fisicamente per aggiornarsi.

## 2.4. - Responsabilità di progetto

Anche se ogni componente ha lo stesso potere decisionale, verranno scelti dei responsabili che cercheranno di non negligenza le parti importanti del progetto.

- Hernest SERANI- Responsabilità per le API, front-end e la comunicazione con la parte back-end
- Enrico BALDASSO- Responsabilità per la parte back-end del applicazione Android
- Stefano CAPPON - Responsabilità per la parte della base di dati, e la gestione del server

A parte le responsabilità individuali, tutti hanno le responsabilità di tenersi aggiornati e collaborare attivamente alla realizzazione progetto, avvisando per tempo eventuali impegni.

# 3. Descrizione dei processi gestionali

## 3.1. - Obiettivi e priorità

Le nostre priorità per il progetto sono:

- Rispettare le consegne dei vari deliverables.
- Creare un'applicazione che possa essere utile anche al di fuori del progetto, specie in un periodo come questo in cui è sempre bene sapere se ci si trova in un luogo affollato.
- Creare un gruppo solido aiutandosi l'un l'altro in caso di bisogno.

## 3.2. - Assunzioni, dipendenze e vincoli

Assumiamo che tutti i membri del gruppo partecipino attivamente al progetto con impegno e disponibilità. Assumiamo che ogni membro abbia delle conoscenze per poter portare a termine l'applicazione. Sarà determinante la velocità di apprendimento dell'utilizzo di Android Studio per sviluppare applicazioni web.



### 3.3. - Gestione dei rischi

Durante la realizzazione del progetto ci saranno dei rischi di cui tenere conto e quindi si cercherà di giocare d'anticipo per evitare e/o controllare tali rischi. Diamo anche una strategia per ciascun rischio che può influire il lavoro del progetto durante lo sviluppo.

N	Descrizione	Probabilità	Impatto	Strategia
1	Perdita o danneggiamento del codice	Bassa	Alto	EVITARE: Usare il servizio hosting di GitHub
2	Incomprensione Interna al gruppo	Media	Medio	MONITORARE: Prendere in considerazione e discutere ogni idea dei membri del team
3	Mancanza di conoscenze, bassa velocità di apprendimento	Bassa	Alto	MITIGARE: Il team cercherà di dare il contributo massimo per la parte di apprendimento
4	Sovrapposizione con altri impegni universitari o impegni di lavoro	Alta	Medio	MITIGARE: Il team coordinerà il tempo di lavoro per il progetto tenendo tutto in considerazione
5	Mancanza della possibilità di testing dell'applicazione	Media	Alto	ACCETTARE: Questo può venire dalla situazione della pandemia Covid19
6	Non rispettare i termini della scadenze	Bassa	Bassa	EVITARE: Hard-working e coordinazione del tempo ed impegno
7	Consegna del progetto con features mancanti	Bassa	Medio	MITIGARE: Fare attenzione e documentare ogni passo dello sviluppo

	Basso	Medio	Alto
Alta		4	
Media		2	5
Bassa	6	7	1-3

### 3.4. - Pianificazione dello staff

Ogni membro del gruppo è tenuto a collaborare con gli altri per le mansioni comuni o se dovesse esserci bisogno di aiuto. Comunque all'interno di una macroarea di lavoro ogni membro avrà una propria mansione.

## **4. Descrizione dei processi tecnici**

### 4.1. - Metodi, strumenti e tecniche

Per la documentazione utilizzeremo una cartella Google Drive condivisa, mentre per le parti di codice per committare utilizzeremo una repository su GitHub.

### 4.2. - Documentazione del software

La documentazione farà riferimento alla data e ai deliverables del progetto:

- Piano di Progetto (18/10/2020)
- Documento di analisi e specifica (27/10/2019)
- Piano di testing (14/11/2019)
- Documento di Progettazione (09/12/2019)
- Codice e messa in linea del sistema (15/01/2021)

Il gruppo svilupperà questi documenti come files condivisi google Drive in modo tale che ogni membro riesca a modificare il documento in contemporanea con gli altri.

### 4.3. - Funzionalità di supporto al progetto

- Mantenere una documentazione sempre aggiornata e priva di elementi superflui.
- Sviluppare l'applicazione, il server e il database tenendo conto che potrebbero essere modificati in fase di progettazione e/o testing.
- Mantenere un buon grado di coordinazione tra i membri del gruppo.

## 5. Pianificazione del lavoro, delle risorse umane e del budget

### 5.1. - WBS - Work Breakdown Structure

1. Piano di progetto
  - 1.1. Definizione obiettivi
  - 1.2. Definizione modello
  - 1.3. Struttura organizzativa
  - 1.4. Individuazione e gestione dei rischi
  - 1.5. Descrizione dei processi tecnici
  - 1.6. Stima dei costi
  - 1.7. Pianificazione del lavoro
    - 1.7.1. WBS
    - 1.7.2. Dipendenze
    - 1.7.3. Risorse necessarie
    - 1.7.4. Allocazione Budget
2. Analisi e specifica
  - 2.1. Analisi di fattibilità
  - 2.2. Definizione dei requisiti
  - 2.3. Analisi dei requisiti funzionali
  - 2.4. Analisi dei requisiti non funzionali
  - 2.5. Analisi dei dati
  - 2.6. Analisi della progettazione grafica
  - 2.7. Stesura documento di analisi e specifica
3. Progettazione
  - 3.1. Server
    - 3.1.1. Scelta Server
    - 3.1.2. Scelta DBMS
  - 3.2. Definizione delle politiche di accesso e sicurezza
  - 3.3. Scelta delle eventuali API
  - 3.4. Pianificazione della interfaccia
  - 3.5. Organizzazione del codice
  - 3.6. Definizione delle strutture dati e classi di supporto
  - 3.7. Pianificazione dei test
4. Sviluppo
  - 4.1. Sviluppo Database nel server
    - 4.1.1. Configurazione Server (1/2 giorno)
    - 4.1.2. Configurazione Database (1/2 giorno)

- 4.1.3. Installazione dipendenze (1/2 giorno)
- 4.2. Sviluppo Applicazione
  - 4.2.1. Front-End
    - 4.2.1.1. Segmentare in blocchi il software da sviluppare (1/2 giorno)
    - 4.2.1.2. Sviluppo Interfaccia (1 mese)
  - 4.2.2. Back-End
    - 4.2.2.1. Segmentare in blocchi il software da sviluppare (1/2 giorno)
    - 4.2.2.2. Implementare Strutture dati di supporto(2 giorni)
    - 4.2.2.3. Implementare Politiche Security(1 giorno)
    - 4.2.2.4. Sviluppo funzionalità (1 mese)
  - 4.2.3. Commento e pulizia del codice
- 5. Test e rilascio
  - 5.1. Test su dispositivi Android
  - 5.2. Risoluzione eventuali bug
  - 5.3. Rilascio su Google Play Store

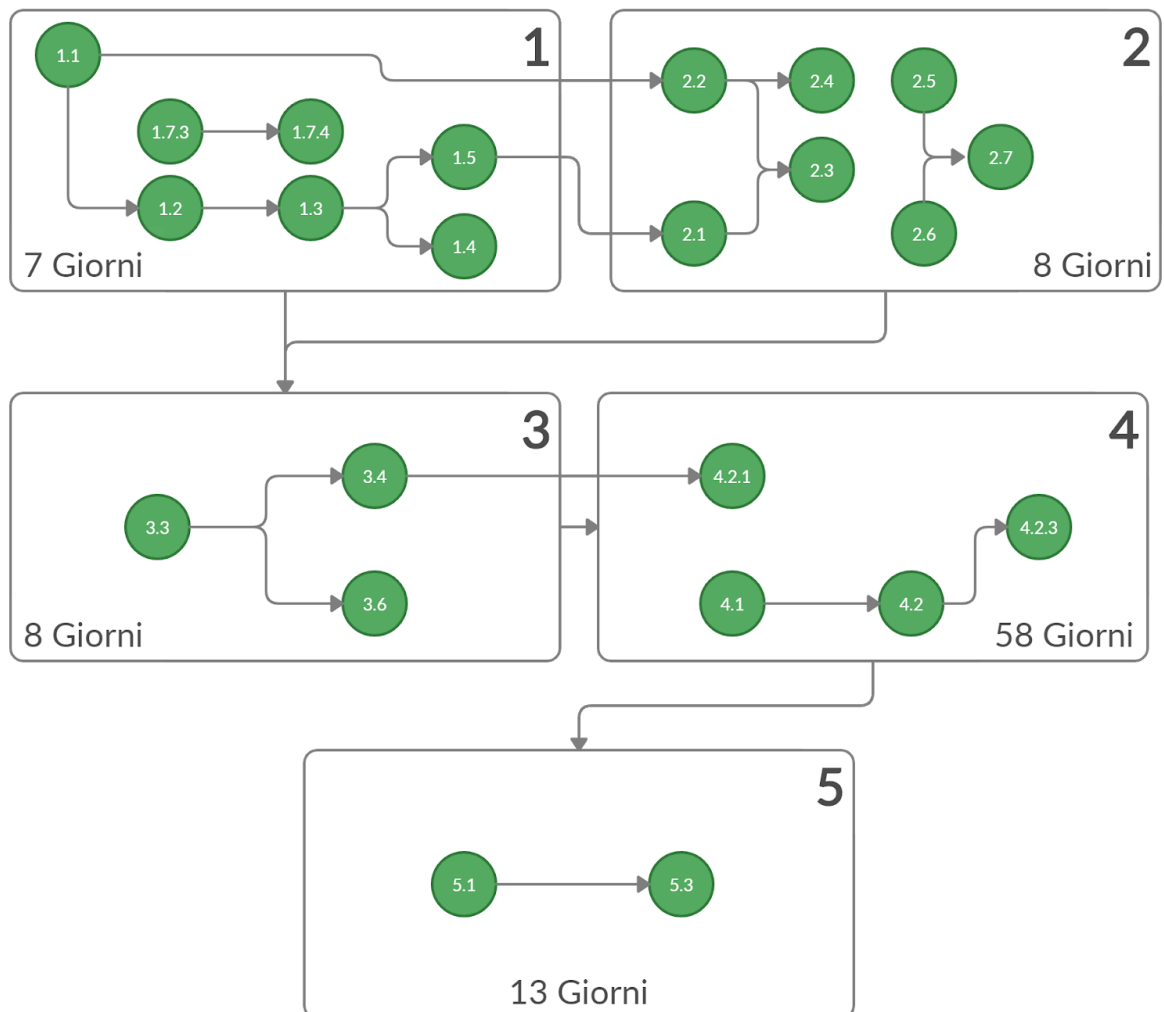
## 5.2. - Dipendenze

N	Fase		Dipendenze	Durata	Data Inizio	Data Fine
1	<b>Piano di Progetto</b>			<b>7 giorni</b>	<b>11/10/2020</b>	<b>18/10/2020</b>
	1.1	Definizione obiettivi		5 giorni	11/10/2020	16/10/2020
	1.2	Definizione modello	1.1			
	1.3	Struttura organizzativa	1.2			
	1.4	Individuazione e gestione dei rischi	1.3			
	1.5	Descrizione dei processi tecnici	1.3			
	1.6	Stima dei costi				
	1.7	Pianificazione del lavoro	1.3	2 giorni	16/10/2020	18/10/2020
		1.7.1 WBS				
		1.7.2 Dipendenze				
		1.7.3 Risorse necessarie				
		1.7.4 Allocazione Budget	1.7.3			

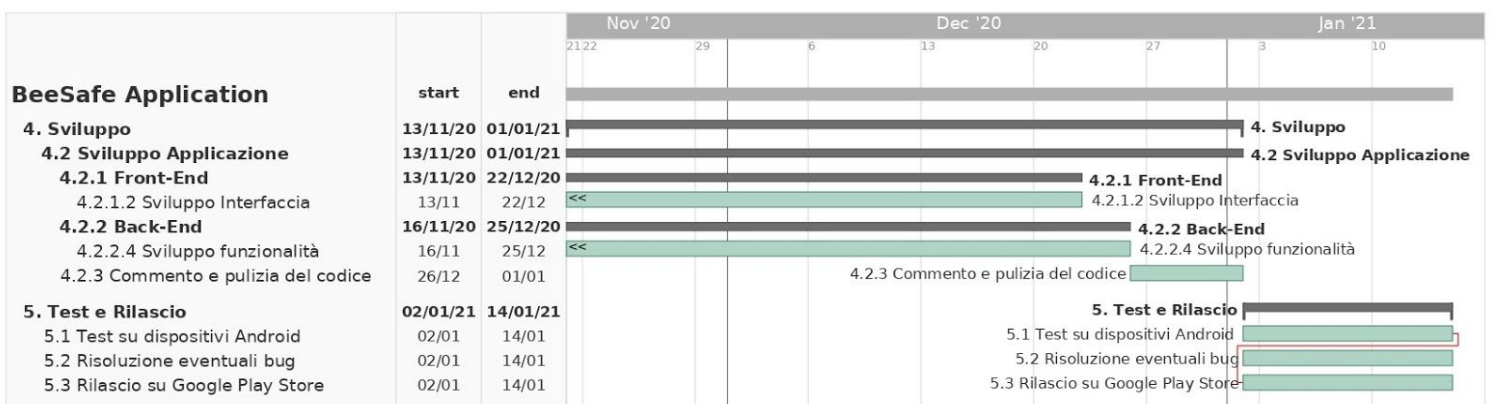
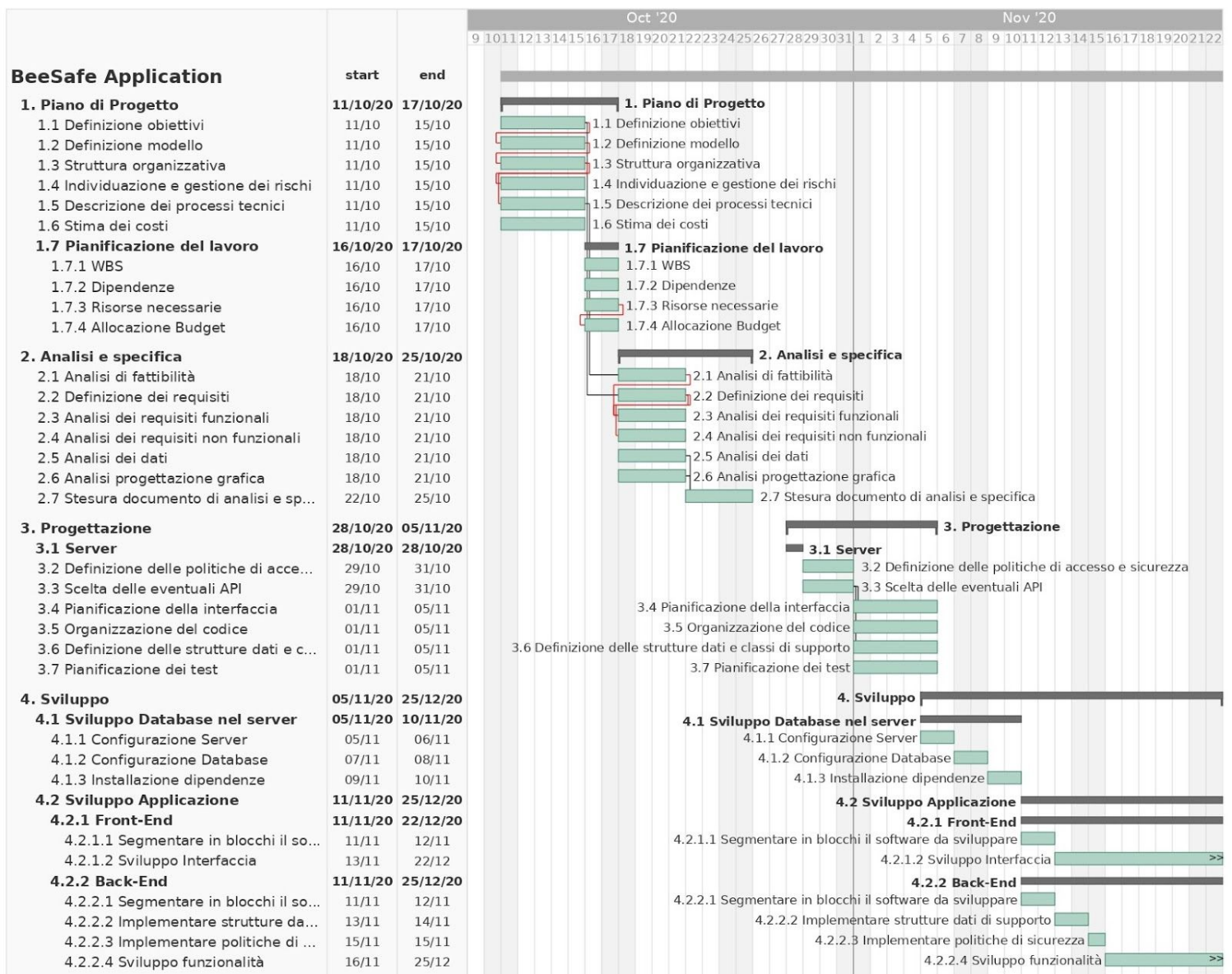
N	Fase			Dipendenze	Durata	Data Inizio	Data Fine
2	Analisi e specifica				8 giorni	18/10/2020	26/10/2020
	2.1	Analisi di fattibilità		1.5	4 giorni	18/10/2020	22/10/2020
	2.2	Definizione dei requisiti		1.1			
	2.3	Analisi dei requisiti funzionali		2.2			
	2.4	Analisi dei requisiti non funzionali		2.2			
	2.5	Analisi dei dati					
	2.6	Analisi progettazione grafica					
	2.7	Stesura documento di analisi e specifica		2.5-2.6	4 giorni	22/10/2020	26/10/2020
N	Fase			Dipendenze	Durata	Data Inizio	Data Fine
3	Progettazione			1 - 2	8 giorni	28/10/2020	05/11/2020
	3.1	Server			1 giorno	28/10/2020	29/10/2020
		3.1.1	Scelta Server				
		3.1.2	Scelta DBMS				
	3.2	Definizione delle politiche di accesso e sicurezza			3 giorni	29/10/2020	01/11/2020
	3.3	Scelta delle eventuali API		2.3			
	3.4	Pianificazione della interfaccia			4 giorni	01/11/2020	05/11/2020
	3.5	Organizzazione del codice					
	3.6	Definizione delle strutture dati e classi di supporto		3.3			
	3.7	Pianificazione dei test		3.6			
N	Fase			Dipendenze	Durata	Data Inizio	Data Fine
4	Sviluppo			2-3.5	58 giorni	05/11/2020	02/01/2020
	4.1	Sviluppo Database nel server		3.1	6 giorni	05/11/2020	11/11/2020
		4.1.1	Configurazione Server	3.1.1	2 giorni	05/11/2020	07/11/2020
		4.1.2	Configurazione Database	3.1.2, 4.1.1	2 giorni	07/11/2020	09/11/2020
		4.1.3	Installazione dipendenze	4.1.2	2 giorni	09/11/2020	11/11/2020
	4.2	Sviluppo Applicazione		4.1	52 giorni	11/11/2020	02/01/2021

		4.2.1	Front-End		3.4	42 giorni	11/11/2020	23/12/2020
			4.2.1.1	Segmentare in blocchi il software da sviluppare		2 giorni	11/11/2020	13/11/2020
			4.2.1.2	Sviluppo Interfaccia		40 giorni	13/11/2020	23/12/2020
		4.2.2	Back-End			45 giorni	11/11/2020	26/12/2020
			4.2.2.1	Segmentare in blocchi il software da sviluppare		2 giorni	11/11/2020	13/11/2020
			4.2.2.2	Implementare strutture dati di supporto		2 giorni	13/11/2020	15/11/2020
			4.2.2.3	Implementare politiche di sicurezza		1 giorni	15/11/2020	16/11/2020
			4.2.2.4	Sviluppo funzionalità		40 giorni	16/11/2020	26/12/2020
		4.2.3	Commento e pulizia del codice		4.2.1, 4.2.2	7 giorni	26/12/2020	02/01/2020
N	Fase				Dipendenze	Durata	Data Inizio	Data Fine
5	Test e rilascio				3.7, 4	13 giorni	02/01/2021	15/01/2021
	5.1	Test su dispositivi Android			4.2.2	13 giorni	02/01/2021	15/01/2021
	5.2	Risoluzione eventuali bug			5.1			
	5.3	Rilascio su Google Play Store			5.1, 5.2			

## Diagramma di Pert



# Diagramma di Gantt





### 5.3. - Risorse necessarie

Le risorse necessarie allo sviluppo del progetto sono:

- Il team di lavoro: noi membri del gruppo
- Hardware: I computer e smartphone dai quali lavoriamo
- Software. Android Studio per lo sviluppo dell'applicazione e Firebase per il server

### 5.4. - Allocazione del budget e delle risorse

Il budget non è rilevante in quanto l'unico fattore di cui tenere conto è il tempo di lavoro dedicato al progetto dai membri del gruppo. A livello di tempo abbiamo stabilito che il progetto durerà fino a metà gennaio, come da scadenza programmata, dove ogni componente indicativamente sarà occupato per un'ora al giorno.

### 5.5. - Pianificazione

Per la pianificazione ci siamo basati con le scadenze che ci sono state assegnate dal committente:

- Piano di Progetto (18/10/2020)
- Documento di analisi e specifica (27/10/2019)
- Piano di testing (14/11/2019)
- Documento di Progettazione (09/12/2019)
- Codice e messa in linea del sistema (15/01/2021)

Oltre a queste date, per le fasi più importanti del progetto verranno inserite delle milestones, ad ogni milestones verrà effettuata una chiamata su discord con tutti i membri del gruppo per aggiornarsi sullo stato di avanzamento del progetto e trovare soluzioni ad eventuali dubbi.