



Progetto di Ingegneria del Software 2025/26

Università Ca' Foscari Venezia

D2: Piano di Progetto

0.3

V.M.F

16/10/2025



Document Informations

NomeProgetto		Acronimo	
Deliverable	Piano di Progetto		
Data di Consegna	21/10/2025		
Team Leader	Francesco Pezzuto	902320@stud.unive.it	
Team members	Matteo Tarushi 900554@stud.unive.it Victor Hortopan 900555@stud.unive.it		

Document History

Version	Issue Date	Stage	Changes	Contributors
0.1	07/ 10 / 2025	Draft	Definizione Gruppo Di Lavoro	FP, MT, VH
0.2	13/ 10/ 2025	Draft	Proposta Iniziale & Coinvolgimento Stackholders	FP, MT, VH
0.3	16/10/2025	Draft	Piano di Progetto	FP, MT, VH
2.0		Final		



Indice

1.Introduzione	6
1.1. Overview del progetto	6
1.2. Deliverables del Progetto	6
1.3. Evoluzione del progetto	7
1.4. Materiali di riferimento	7
1.5. Definizioni e Abbreviazioni	7
2.Organizzazione del progetto	10
2.1 Modello del processo	10
2.1.2 Struttura temporale del processo Agile adottato	10
2.1.3 Flusso di lavoro Agile adottato per GreenZone	11
2.2 Struttura Organizzativa	11



2.2.1 Motivazioni della scelta	12
2.2.2 Struttura organizzativa a matrice del Team	12
2.2.3 Tipologia di Team	12
2.2.4 Schema Team Controllato Decentralizzato	13
2.3 Interfacce Organizzative	13
2.4 Responsabilità di Progetto	14
2.4.1 Ruoli e incarichi principali	14
2.4.2 Modalità operative	15
3.Processi Gestionali	16
3.1 Obiettivi e Priorità	16
3.2 Assunzioni, Dipendenze e Vincoli	17
3.2.1 Assunzioni	17
3.2.2 Dipendenze	17
3.2.3 Vincoli	17
3.3 Gestione dei Rischi	18
3.3.1 Tabella dei rischi	19
3.3.2 Modello dei rischi	20
3.4 Meccanismi di Controllo	20
3.5 Pianificazione dello Staff	21
4.Processi Tecnici	22
4.1 Metodi, Strumenti e Tecniche	22
4.1.1 Strumenti	22
4.1.2 Metodi di sviluppo	22
4.1.3 Tecniche	23
4.2 Documentazione	23
4.3 Funzionalità di supporto al progetto	23



5. Pianificazione delle attività e delle risorse	24
5.1 Work Breakdown Structure	24
5.1.1 Funzioni principali	24
5.1.2 Attività	25
5.1.3 Milestone principali	26
5.1.4 WBS	27
5.2 Dipendenze	27
5.2.1 Tabella dipendenze	28
5.3 Risorse necessarie	29
5.3.1 Diagramma di PERT	29
5.3.2 Stima dei costi	29
5.4 Allocazione del Budget e delle risorse	30
5.4.1 Deadlines e Milestone	30
5.4.2 Diagramma di Gantt	30
5.4.3 Scheduling di Progetto	31



1.Introduzione

1.1. Overview del progetto

Il progetto consiste nello sviluppo di un'applicazione mobile, **GreenZone**, il cui obiettivo è quello di facilitare la scoperta e la promozione di luoghi sostenibili nella propria area (come punti di ricarica elettrica, fontanelle, aree verdi, negozi bio).

L'app offre una mappa interattiva che consente di:

- Visualizzare le diverse categorie di luoghi sostenibili;
- Calcolare percorsi a piedi o in bici verso tali punti;
- Filtrare i risultati per categoria o distanza;
- Segnalare nuovi luoghi o correggere informazioni errate;
- Salvare i luoghi preferiti per un accesso rapido

L'obiettivo principale è favorire comportamenti sostenibili nella vita quotidiana, rendendo più semplice per gli utenti contribuire al benessere ambientale della propria comunità.

1.2. Deliverables del Progetto

Le scadenze per la creazione del progetto sono le seguenti:

- D0: Definizione dei gruppi di lavoro (7/10/2025)
- D1: Proposta iniziale (14/10/2025)
- D2: Piano del progetto (21/10/2025)
- D3: Documento dei Requisiti (11/11/2025)
- D4: Piano di Testing (18/11/2025)
- D5: Documento di Progettazione (30/11/2025)
- D6: Versione 1.0 del codice sorgente (22/12/2025)
- D7: Versione 1.1 del codice e allineamento documentazione (26/01/2026)



1.3. Evoluzione del progetto

Il progetto è stato concepito in modo scalabile e modulare, così da consentire in futuro l'aggiunta di nuove funzionalità e l'ampliamento delle categorie di luoghi sostenibili presenti all'interno dell'applicazione.

Tra le possibili evoluzioni, si ipotizza l'integrazione con eventi locali sostenibili, come mercatini, giornate ecologiche o iniziative di riciclo promosse dai quartieri o dalle amministrazioni locali.

Queste espansioni permetterebbero di trasformare GreenZone in una piattaforma non solo informativa, ma anche partecipativa e comunitaria, capace di mettere in contatto cittadini e realtà territoriali impegnate nella sostenibilità.

Tuttavia, poiché il progetto si trova ancora nelle fasi iniziali di sviluppo, non è possibile definire con certezza quali funzionalità aggiuntive verranno effettivamente implementate rispetto a quanto previsto nella proposta iniziale. Le evoluzioni future dipenderanno dai risultati della prima versione e dalle risorse disponibili.

1.4. Materiali di riferimento

Durante lo sviluppo di **GreenZone**, il team farà riferimento ai seguenti materiali:

- Slide del corso di **Ingegneria del Software**;
- Documentazione ufficiale di **Java** per la programmazione dell'applicazione;
- Documentazione ufficiale di **Android Studio** e delle **API Android** per la progettazione dell'interfaccia utente e la gestione delle funzionalità native;
- Documentazione **Google Maps API** per l'integrazione della mappa e la gestione dei marker geolocalizzati;
- Documentazione ufficiale di **Firebase** per il back-end e la gestione del database;
- Linee guida **Google** per la progettazione UX/UI;
- Repository **GitHub** condiviso per il versionamento e la collaborazione;
- **Google Drive** per la condivisione e l'archiviazione dei documenti tra il team;

1.5. Definizioni e Abbreviazioni

- [App](#)



Applicazione mobile sviluppata per dispositivi Android.

- **[Java](#)**

Linguaggio di programmazione orientato agli oggetti utilizzato per lo sviluppo dell'app.

- **[Android Studio](#)**

Ambiente di sviluppo integrato (IDE) ufficiale per la creazione di applicazioni Android.

- **[API](#)**

Application Programming Interface, insieme di funzioni e procedure che permettono l'interazione tra software diversi.

- **[Google Maps API](#)**

Servizi forniti da Google per l'integrazione di mappe interattive e funzionalità di geolocalizzazione.

- **[Front-end](#)**

Parte visibile dell'applicazione con cui interagisce direttamente l'utente.

- **[Back-end](#)**

Parte logica e di gestione dati dell'applicazione, che elabora le richieste del front-end e comunica con il Database.

- **[Database](#)**

Sistema di archiviazione strutturata dei dati relativi ai luoghi sostenibili, agli utenti e alle segnalazioni (nel nostro caso, viene gestito tramite Firebase).

- **[Firebase](#)**

Piattaforma Google utilizzata per la gestione del back-end, del database in tempo reale e dell'autenticazione degli utenti.

- **[UI/UX](#)**

Interfaccia Utente (User Interface) e Esperienza Utente (User Experience), aspetti fondamentali per garantire usabilità e chiarezza dell'app.

- **[GitHub](#)**



Piattaforma online per il versionamento del codice e la collaborazione tra sviluppatori.

- [Google Drive](#)

Servizio cloud di Google utilizzato per la condivisione e l'archiviazione della documentazione di progetto.

- [Excel](#)

Software di Microsoft Office utilizzato per l'analisi dei dati raccolti dai questionari e la creazione di grafici e tabelle di sintesi.

- [Word](#)

Software di Microsoft Office utilizzato per creare, modificare e formattare documenti come lettere, relazioni, libri, ecc.

- [Whatsapp](#)

Applicazione di messaggistica utilizzata dal team per la comunicazione e il coordinamento rapido delle attività.

- [Discord](#)

Applicazione social di messaggistica istantanea tramite chiamate vocali, videochiamate, messaggi di testo e contenuti multimediali

- [Google form](#)

Strumento di Google per la creazione e la distribuzione di questionari, utilizzato per la raccolta dei feedback dagli utenti e stakeholder.



2. Organizzazione del progetto

2.1 Modello del processo

Il modello di sviluppo scelto per GreenZone è l'Agile, in quanto consente di ottenere risultati di qualità in tempi ridotti e con maggiore flessibilità.

Il modello Agile è stato preferito rispetto al modello Waterfall per i seguenti motivi:

- Maggiore flessibilità in caso di modifiche dei requisiti.
- Rilascio di versioni intermedie funzionanti nell'app.
- Coinvolgimento continuo del docente e degli utenti finali nei processi decisionali.
- Possibilità di valutare e migliorare ad ogni sprint il lavoro svolto.

Questo approccio è ideale per progetti come GreenZone che prevedono rilasci progressivi, feedback continui dagli utenti e adattamenti incrementali delle funzionalità.

2.1.2 Struttura temporale del processo Agile adottato

Il progetto è stato suddiviso in sprint di 2 settimane, per una durata complessiva stimata di 12 settimane (6 sprint).

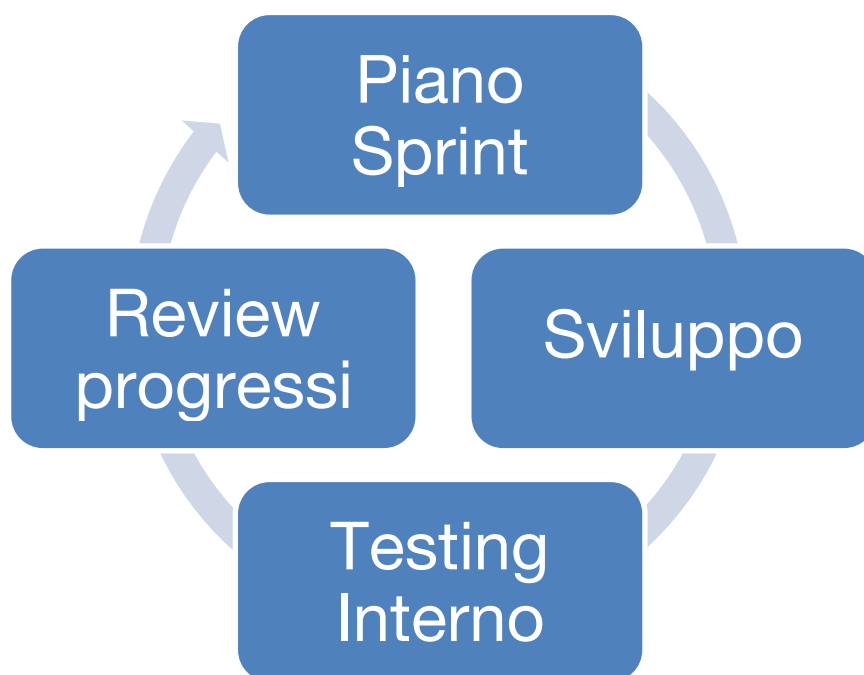
Ogni sprint prevede attività di pianificazione, sviluppo, test e revisione, come riportato di seguito:

Attività	Durata stimata	Descrizione
Piano Sprint	1 giorno	Definizione degli obiettivi dello sprint, analisi delle priorità e assegnazione dei compiti al team.
Sviluppo	8-9 giorni	Sviluppo delle funzionalità pianificate



Testing interno	2 giorni	Verifica del funzionamento, debug e controllo
Review progressi	1 giorno	Review del lavoro completato, discussione di miglioramenti e pianificazione successiva

2.1.3 Flusso di lavoro Agile adottato per GreenZone



2.2 Struttura Organizzativa

La struttura organizzativa adottata per GreenZone è a matrice, in cui i membri del team collaborano in modo orizzontale, pur mantenendo un coordinamento centrale attraverso la figura del Team Leader/ Product Owner (nel nostro caso rappresentato da un'unica persona).



2.2.1 Motivazioni della scelta

Favorisce la comunicazione diretta e la collaborazione trasversale tra i ruoli. Consente una gestione efficiente delle risorse su più attività contemporaneamente ed è coerente con la metodologia Agile che privilegia i team auto-organizzati e flessibili.

2.2.2 Struttura organizzativa a matrice del Team

Team Leader/Product Owner (Coordina, pianifica e comunica con il docente)				
Funzioni	Front-end Dev	Back-end Dev	Documentazione	Testing/Debugging
Analisi dei requisiti	Analisi	-	Redazione	-
Progettazione	Mockup interfacce	Struttura DB	Documentazione	Test preliminare
Sviluppo	Implementazione	Funzionalità	Supporto e stesura	Verifica funzionalità
Testing finale	Ottimizzazione	Integrazioni	Revisione Documentazione	Test finale

2.2.3 Tipologia di Team

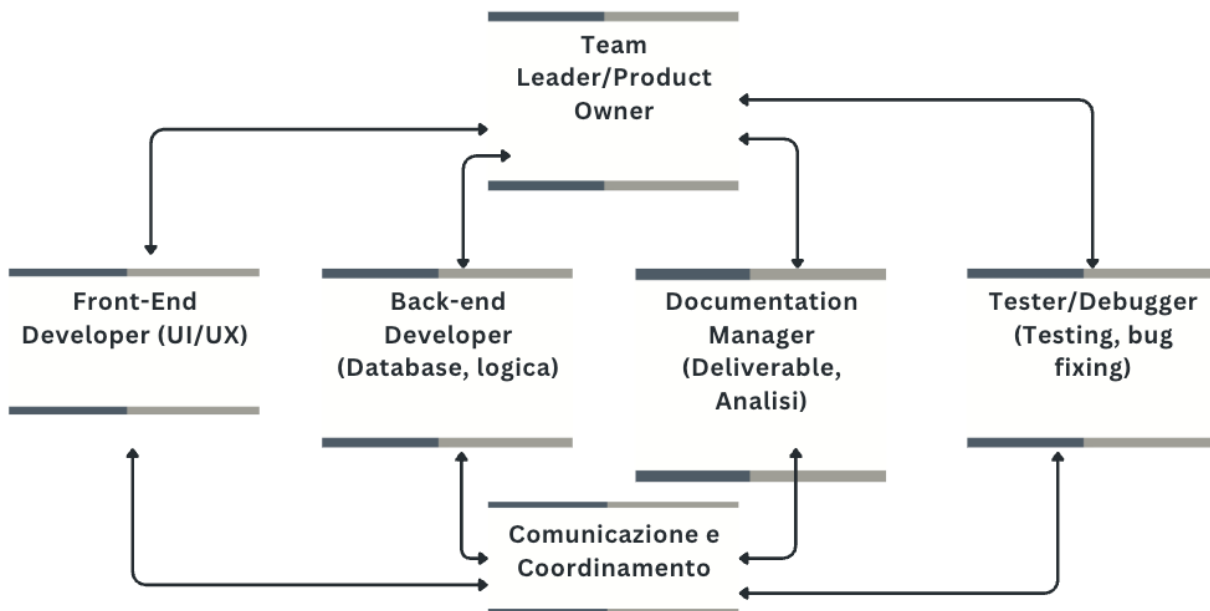
Il team GreenZone è di tipo controllato decentralizzato, perché garantisce un buon equilibrio tra autonomia operativa e coordinamento centrale, ideale per il nostro team dove:

- È presente un leader che coordina il lavoro e definisce le priorità.
- I membri hanno autonomia decisionale nelle proprie aree di competenza.
- La comunicazione è orizzontale tra i membri e verticale con il leader.



- La risoluzione dei problemi è collettiva, ma l'esecuzione delle attività è distribuita.

2.2.4 Schema Team Controllato Decentralizzato



2.3 Interfacce Organizzative

Durante lo sviluppo di GreenZone è prevista la comunicazione costante sia all'interno del team sia con le entità esterne coinvolte nel progetto. Una gestione efficace delle interfacce organizzative consente di mantenere un flusso informativo chiaro, tempestivo e coerente con gli obiettivi del progetto.

- **Comunicazione interna:**

Tutti i membri del gruppo partecipano attivamente alle attività di sviluppo, documentazione e verifica. Le comunicazioni avverranno in modo orizzontale, favorendo la collaborazione e la condivisione continua di idee e aggiornamenti sullo stato del progetto.

Per garantire tutto ciò utilizzeremo degli strumenti di comunicazione come Whatsapp e Discord per comunicazioni rapide e coordinamento quotidiano.

Inoltre, grazie alle competenze comunicative e relazionali evidenziate nei CV dei membri del team, il gruppo dispone di solide capacità di collaborazione e lavoro di



squadra, elementi che consentono di mantenere un ambiente di lavoro unito e produttivo.

- **Comunicazione esterna:**

Oltre alla comunicazione interna, il gruppo si interfacerà periodicamente con figure esterne al team di sviluppo, ovvero il professore **A. Cortesi**, che sarà tenuto aggiornato sull'avanzamento del progetto e sulla consegna dei deliverables, con particolare attenzione alla qualità dei documenti e alla coerenza dei dati.

Il professore **A.Spanò**, che sarà consultato per questioni tecniche riguardanti l'ambiente di sviluppo, le API, la gestione del database e altre problematiche legate al codice e alle tecnologie adottate.

Infine, durante le fasi finali del progetto, verranno coinvolti alcuni **tester** per fornire feedback sull'esperienza utente, sulle funzionalità dell'app e sulla stabilità di sistema.

2.4 Responsabilità di Progetto

Il team adotta una struttura organizzativa a controllo decentralizzato, nelle quali ogni membro assume ruoli specifici ma collabora in modo attivo e trasversale alle varie fasi del progetto.

Questo approccio consente una distribuzione equilibrata delle responsabilità, un miglior coordinamento e un efficace monitoraggio delle attività in stile Agile.

2.4.1 Ruoli e incarichi principali

Ruolo	Membro	Responsabilità Principali
Team Leader/Product Owner/ Documentation Supervisor/Tester	Pezzuto Francesco	Coordina le attività del gruppo, definisce le priorità degli sprint e mantiene la comunicazione con i docenti. Supervisiona il codice e la redazione e revisione dei documenti riguardanti i deliverable.
Back-end Developer/System Administrator/Tester	Hortopan Victor	Si occupa dello sviluppo del back-end, della gestione del database, delle API e dell'integrazione con il front-end. Configura e gestisce



		l'ambiente di sviluppo e testing ed esegue i test funzionali.
UI Designer/Tester/Back-end Developer	Tarushi Matteo	Responsabile della parte visiva dell'app. Esegue test di usabilità, controlla la qualità complessiva del prodotto e integra lo sviluppo del back-end

2.4.2 Modalità operative

Le attività vengono pianificate all'inizio di ogni sprint, tenendo conto delle competenze specifiche di ciascun membro del team.

Tutti i membri possono supportarsi reciprocamente nelle varie fasi del progetto (sviluppo, test o documentazione) per garantire la continuità del lavoro e rispettare le scadenze.

Al termine di ogni sprint si tiene una riunione di revisione per valutare i risultati, raccogliere feedback e definire le priorità per lo sprint successivo.



3.Processi Gestionali

3.1 Obiettivi e Priorità

L'obiettivo principale del progetto GreenZone è lo sviluppo di un'applicazione mobile che promuova comportamenti sostenibili e consenta agli utenti di monitorare, condividere e migliorare le proprie abitudini ecologiche quotidiane.

L'app mira a fornire uno strumento intuitivo, educativo e socialmente utile, che favorisca la sensibilizzazione ambientale e la creazione di una community eco-consapevole.

Gli obiettivi specifici sono:

- **Realizzare un'app stabile e user-friendly per dispositivi Andorid.**
- **Integrare funzionalità di monitoraggio delle azioni ecologiche**
- **Offrire un sistema di incentivazione alla partecipazione**
- **Garantire la sicurezza e l'affidabilità dei dati utente**
- **Rispettare i tempi di consegna e i deliverable previsti**

Le priorità:

- **Rispettare le scadenze:**

Garantire la consegna puntuale dei deliverable per ogni fase del progetto.

- **Qualità del software:**

Mantenere un elevato livello qualitativo sia nel codice che nell'esperienza utente.

- **Usabilità e accessibilità:**

Assicurare un'interfaccia semplice, chiara e accessibile a tutti gli utenti.

- **Sostenibilità e innovazione:**

Sviluppare funzionalità coerenti con la missione ecologica del progetto.

- **Collaborazione efficace:**

Favorire il dialogo costante e la cooperazione tra i membri del team.



3.2 Assunzioni, Dipendenze e Vincoli

3.2.1 Assunzioni

- Tutti i membri del team dispongono dell'hardware necessario.
- Gli strumenti software utilizzati (Android Studio, Github, Google Drive, ecc.) saranno pienamente funzionanti per tutta la durata del progetto.
- Le competenze dichiarate nei CV (comunicazione, problem solving, ecc.) saranno mantenute e valorizzate nel lavoro di gruppo.
- Gli utenti tester forniranno feedback in tempi utili per l'analisi e la correzione.

3.2.2 Dipendenze

- L'integrazione tra front-end e back-end dipende dal completamento e dal corretto funzionamento delle API.
- La fase di testing dipende dallo sviluppo completo delle funzionalità principali.
- La redazione dei documenti è vincolata al rispetto delle scadenze prefissate.

3.2.3 Vincoli

- **Temporalì:**

Rispetto delle date ufficiali di consegna indicate nel piano di corso.

- **Tecnici:**

Utilizzo di linguaggi e strumenti specificati nel progetto.

- **Organizzativi:**

Disponibilità dei membri secondo gli orari stabiliti per gli sprint settimanali.

- **Qualitativi:**

Aderenza agli standard di qualità definiti e ai requisiti concordati con i docenti.



3.3 Gestione dei Rischi

Durante la fase di pianificazione, il team ha identificato e analizzato i potenziali **rischi** che potrebbero influire **negativamente** sullo svolgimento del progetto.

L'obiettivo di questa attività è quello di **prevedere, monitorare e mitigare** gli eventi che potrebbero compromettere tempi, qualità o risultati del prodotto finale.

La gestione dei rischi avviene secondo un approccio **proattivo e continuo**, coerente con la metodologia **Agile**.

In particolare:

- Ogni rischio viene valutato in base alla **probabilità di accadimento** e all'**impatto** sul progetto.
- Durante ogni **sprint**, il team verifica se si siano verificati nuovi rischi o variazioni di gravità rispetto a quelli già individuati.
- Le **azioni** vengono definite e applicate tempestivamente, con l'obiettivo di ridurre al minimo gli effetti negativi.
- I rischi principali sono stati classificati in tre categorie:
 - **Organizzativi**: legati a ritardi, assenze o scarsa comunicazione tra i membri.
 - **Tecnici**: derivanti da errori di sviluppo, problemi di integrazione o malfunzionamenti.
 - **Gestionali**: collegati a una pianificazione inadeguata, carichi di lavoro errati o perdita di dati.

La seguente **tabella** riassume i principali rischi individuati, con l'indicazione della loro probabilità, dell'impatto e delle misure correttive previste.



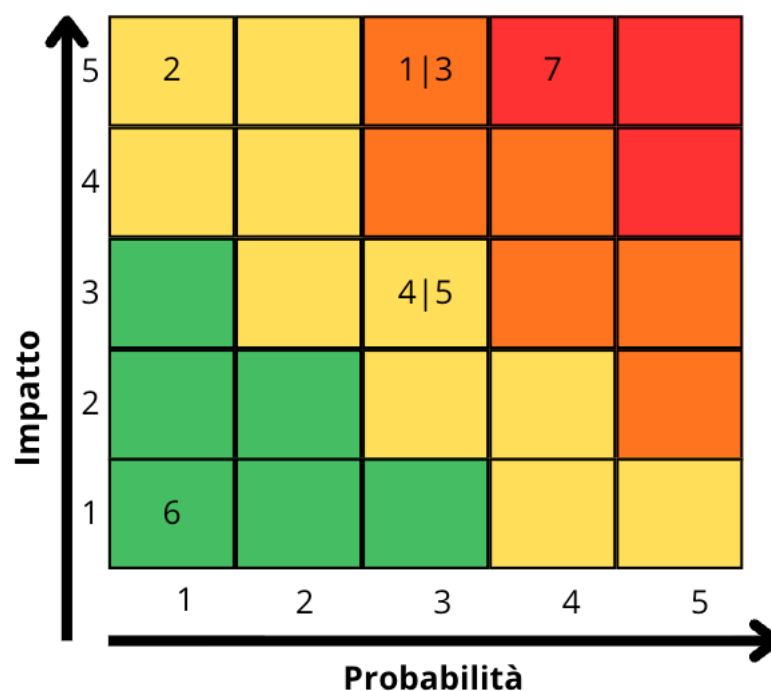
3.3.1 Tabella dei rischi

ID	Rischio	Probabilità	Impatto	Strategia di azione
1	Assenza o ritardo di un membro del team	3	5	Pianificazione flessibile e supporto incrociato tra i membri
2	Malfunzionamento o perdita di dati	1	5	Backup regolari su Github e Drive, salvataggi locali
3	Errori di integrazione front-end/back-end	3	5	Testing frequente e revisione del codice condiviso
4	Sovraccarico di lavoro o gestione del tempo inadeguata	3	3	Pianificazione settimanale realistica e divisione equa dei task
5	Difficoltà tecniche con strumenti o framework	3	3	Consultazione documentazione ufficiale e supporto docente tecnico
6	Feedback tardivo dai tester	3	1	Anticipare la fase di testing interno e predisporre sessioni extra di verifica
7	Bug critici non rilevati	4	5	Testing incrementale e



				controllo manuale dopo ogni sprint
--	--	--	--	---------------------------------------

3.3.2 Modello dei rischi



Legenda:

- 1 - probabilità molto bassa
- 2 - probabilità bassa
- 3 - probabilità media
- 4 - probabilità alta
- 5 - probabilità molto alta

3.4 Meccanismi di Controllo

Per garantire un monitoraggio costante dell'avanzamento del progetto, verranno adottati i seguenti strumenti e procedure:



- **Riunioni settimanali:** pianificate su Whatsapp ed effettuate tramite Discord per discutere dell'avanzamento, criticità e obiettivi del prossimo sprint.
- **Versionamento su GitHub:** monitoraggio dei commit e revisione dei branch da parte del Team Leader.
- **Documentazione condivisa:** tutti i file ufficiali (deliverable, relazioni, diagrammi) saranno archiviati su Google Drive in cartelle strutturate.
- **Verifiche dei deliverables:** controllo formale dello stato del progetto a ogni consegna D1–D7 per garantire coerenza tra codice e documentazione.

3.5 Pianificazione dello Staff

Per poter sviluppare adeguatamente il progetto GreenZone, è necessario disporre di un insieme di competenze tecniche e organizzative che consentano al team di affrontare in modo efficiente tutte le fasi del ciclo di vita del software. Alcune di queste conoscenze risultano già acquisite dai membri del gruppo, mentre altre verranno consolidate o sviluppate nel corso del progetto attraverso la pratica e l'approfondimento.

Tra le competenze pregresse ritenute fondamentali figurano la conoscenza del linguaggio di programmazione **Java**, che rappresenta la base per la realizzazione dell'applicazione mobile, e la capacità di comprendere e gestire la **struttura del database Firebase**, utilizzata per la memorizzazione dei dati e la sincronizzazione in tempo reale.

È inoltre richiesta familiarità con l'ambiente di sviluppo **Android Studio**, strumento principale per la progettazione, l'implementazione e il testing dell'applicazione. L'utilizzo di **GitHub** costituisce un ulteriore requisito indispensabile: ogni membro lavora sul proprio branch dedicato, contribuendo poi all'integrazione nel master branch comune a tutto il gruppo, garantendo così una gestione ordinata del codice e un efficace controllo di versione.

Alcune di queste competenze, come l'utilizzo avanzato dell'ambiente di sviluppo Android e la gestione collaborativa del codice tramite **Git**, verranno progressivamente perfezionate durante lo svolgimento del progetto, grazie al lavoro di gruppo e al confronto continuo tra i membri. In parallelo, saranno sviluppate ulteriormente anche le capacità trasversali come la comunicazione, problem solving e pianificazione, già evidenziate nei **CV** dei componenti del team, le quali contribuiranno a mantenere un ambiente di lavoro cooperativo, motivato e orientato al raggiungimento degli obiettivi comuni.



4. Processi Tecnici

4.1 Metodi, Strumenti e Tecniche

4.1.1 Strumenti

Lo sviluppo del progetto GreenZone richiede l'utilizzo di un ambiente dedicato alla creazione di applicazioni per la piattaforma Android.

Per tale necessità si è scelto di utilizzare **Android Studio**, un ambiente di sviluppo integrato (IDE) che consente la realizzazione completa dell'applicazione, dalla progettazione dell'interfaccia grafica alla scrittura del codice, fino alle fasi di testing e debugging. Android Studio è disponibile per tutti i principali sistemi operativi desktop (Windows, macOS e Linux) e permette di **virtualizzare dispositivi Android**, consentendo di testare l'applicazione in modo rapido e sicuro anche ai membri del team che non possiedono uno smartphone Android.

Per garantire un testing più realistico e verificare il comportamento dell'app su diversi dispositivi e versioni del sistema operativo, il gruppo ha comunque ritenuto necessario effettuare prove anche su **dispositivi fisici reali**, utilizzando smartphone secondari. Ciò consente di evitare possibili incongruenze dovute alla differenza tra ambienti virtuali e fisici, migliorando così l'affidabilità complessiva dei test.

4.1.2 Metodi di sviluppo

Come metodo di sviluppo, il team ha deciso di suddividere il progetto in **diversi task** assegnati ai membri del gruppo in base alle rispettive competenze.

Le attività vengono organizzate in **sprint settimanali**, durante i quali ogni componente si dedica al completamento dei propri compiti. Al termine di ciascuna settimana, vengono fissate **scadenze intermedie** per monitorare l'avanzamento del lavoro e verificare il rispetto della pianificazione.

In caso di difficoltà tecniche o ritardi imprevisti, il gruppo adotta un approccio **collaborativo e flessibile**, che prevede momenti di confronto e supporto reciproco per la risoluzione dei problemi.

Qualora un membro completi le proprie attività in anticipo, potrà fornire assistenza ai compagni che riscontrano maggiori complessità, con l'obiettivo di mantenere un ritmo di sviluppo equilibrato e di garantire la continuità del lavoro.

Questo metodo di lavoro, ispirato ai principi della **metodologia Agile**, consente di migliorare la comunicazione interna, aumentare la produttività del team e favorire un costante controllo sull'andamento del progetto.



4.1.3 Tecniche

Dal punto di vista tecnico, il gruppo ha deciso di utilizzare **Java** come linguaggio di programmazione principale, in quanto rappresenta la soluzione più stabile e compatibile per lo sviluppo di applicazioni Android.

L'attività di programmazione sarà supportata dalla **documentazione ufficiale Android**, che fornisce linee guida e buone pratiche per la progettazione di applicazioni efficienti, sicure e scalabili.

4.2 Documentazione

La documentazione del progetto **GreenZone** viene redatta parallelamente alle attività di sviluppo, in modo da garantire coerenza e aggiornamento costante tra la parte implementativa e quella descrittiva.

Nel corso del progetto verranno prodotti i seguenti documenti principali:

- **Piano di Progetto (D2):** descrive la pianificazione temporale, organizzativa e tecnica del progetto.
- **Documento dei Requisiti (D3):** raccoglie e specifica i requisiti funzionali e non funzionali.
- **Piano di Testing (D4):** definisce le strategie di verifica, i test case e i criteri di validazione.
- **Documento di Progettazione (D5):** illustra l'architettura del sistema, le scelte tecnologiche e la struttura del database.

Tutti i documenti vengono redatti in formato digitale, utilizzando **Word** e condivisi tramite **Google Drive**, che consente la scrittura simultanea e la gestione collaborativa delle revisioni.

Le versioni definitive dei documenti saranno consegnate secondo il calendario stabilito per le diverse deliverables (D1–D7).

4.3 Funzionalità di supporto al progetto

Lo sviluppo di un software di qualità rappresenta uno degli obiettivi principali del gruppo **GreenZone**.

Per raggiungere questo traguardo, il team si impegna a:

1. Mantenere costantemente aggiornata la documentazione tecnica e progettuale.
2. Scrivere codice **efficiente** in termini di tempo di esecuzione e consumo di risorse, considerando che l'applicazione dovrà essere eseguita su dispositivi mobili con capacità computazionale inferiore rispetto ai calcolatori desktop.



3. Effettuare **revisioni periodiche del codice** e test ciclici per individuare e correggere tempestivamente eventuali bug o criticità che possano compromettere la qualità o l'esperienza utente.

Per il supporto tecnico e gestionale, il gruppo utilizza i seguenti strumenti:

- **GitHub:** servizio per la gestione del codice sorgente, il versionamento e la collaborazione in tempo reale. Permette di mantenere la cronologia delle modifiche e sincronizzare il lavoro tra tutti i membri del team.
- **Google Drive:** piattaforma cloud utilizzata per la condivisione dei documenti, dei file di progetto e delle risorse utili.
- **Documenti Word:** strumento per la redazione collaborativa dei deliverable, con possibilità di scrittura simultanea e revisione condivisa.
- **Android Studio:** ambiente di sviluppo integrato utilizzato per la realizzazione, il debugging e il testing dell'applicazione.
- **Discord e WhatsApp:** piattaforme di comunicazione istantanea per il coordinamento quotidiano, la condivisione di aggiornamenti e la gestione delle riunioni di gruppo.

5. Pianificazione delle attività e delle risorse

5.1 Work Breakdown Structure

La Work Breakdown Structure rappresenta la scomposizione gerarchica del progetto GreenZone in funzioni, attività e task elementari. Tale suddivisione è stata progettata in modo coerente con il modello di processo **Agile**, consentendo una pianificazione iterativa e incrementale, basata su cicli di sviluppo (sprint) della durata di due settimane ciascuno.

5.1.1 Funzioni principali

- **F1 – Analisi e impostazione del progetto:** pianificazione delle attività, organizzazione delle riunioni settimanali, verifica degli sprint e aggiornamento di esso.
- **F2 – Architettura e prototipo grafico:** Creazione della struttura del database e prime interfacce UI.
- **F3 – Implementazione funzionalità principali:** Sviluppo dell'applicazione e implementazione back-end, versione prototipo funzionante.
- **F4 – Integrazione e test intermedio:** Collegamento front-end e back-end con test funzionali.
- **F5 – Ottimizzazione e test finale:** esecuzione dei test funzionali e strutturali, verifica delle prestazioni, individuazione e correzione dei bug.



- **F6 – Revisione e rilascio finale:** Revisione documenti e pubblicazione definitiva.

5.1.2 Attività

- **A1 (Analisi e impostazione del progetto):**
 - **A1.1:** Raccolta e definizione dei requisiti funzionali e non funzionali
 - **A1.2:** Analisi dei vincoli tecnici e di contesto
 - **A1.3:** Stesura del Piano di Progetto
 - **A1.4:** Validazione iniziale del progetto
- **A2 (Architettura e prototipo grafico):**
 - **A2.1:** Definizione dell'architettura software e del flusso dei dati
 - **A2.2:** Creazione dei mockup e della struttura grafica dell'app
 - **A2.3:** Progettazione e realizzazione del database su Firebase
 - **A2.4:** Revisione documentazione tecnica intermedia
- **A3 (Implementazione funzionalità principali):**
 - **A3.1:** Sviluppo front-end (interfacce utenti principali)
 - **A3.2:** Sviluppo back-end
 - **A3.3:** Integrazione con Firebase e test parziali
 - **A3.4:** Aggiornamento del Piano di Testing
- **A4 (Integrazione e test intermedio):**
 - **A4.1:** Collegamento completo tra front-end e back-end
 - **A4.2:** Test di compatibilità e verifica funzionale
 - **A4.3:** Allineamento con il Documento di Progettazione
- **A5 (Ottimizzazione e test finale):**
 - **A5.1:** Ottimizzazione del codice e delle prestazioni
 - **A5.2:** Test di usabilità e validazione con utenti esterni
 - **A5.3:** Correzione bug segnalati e miglioramento UI

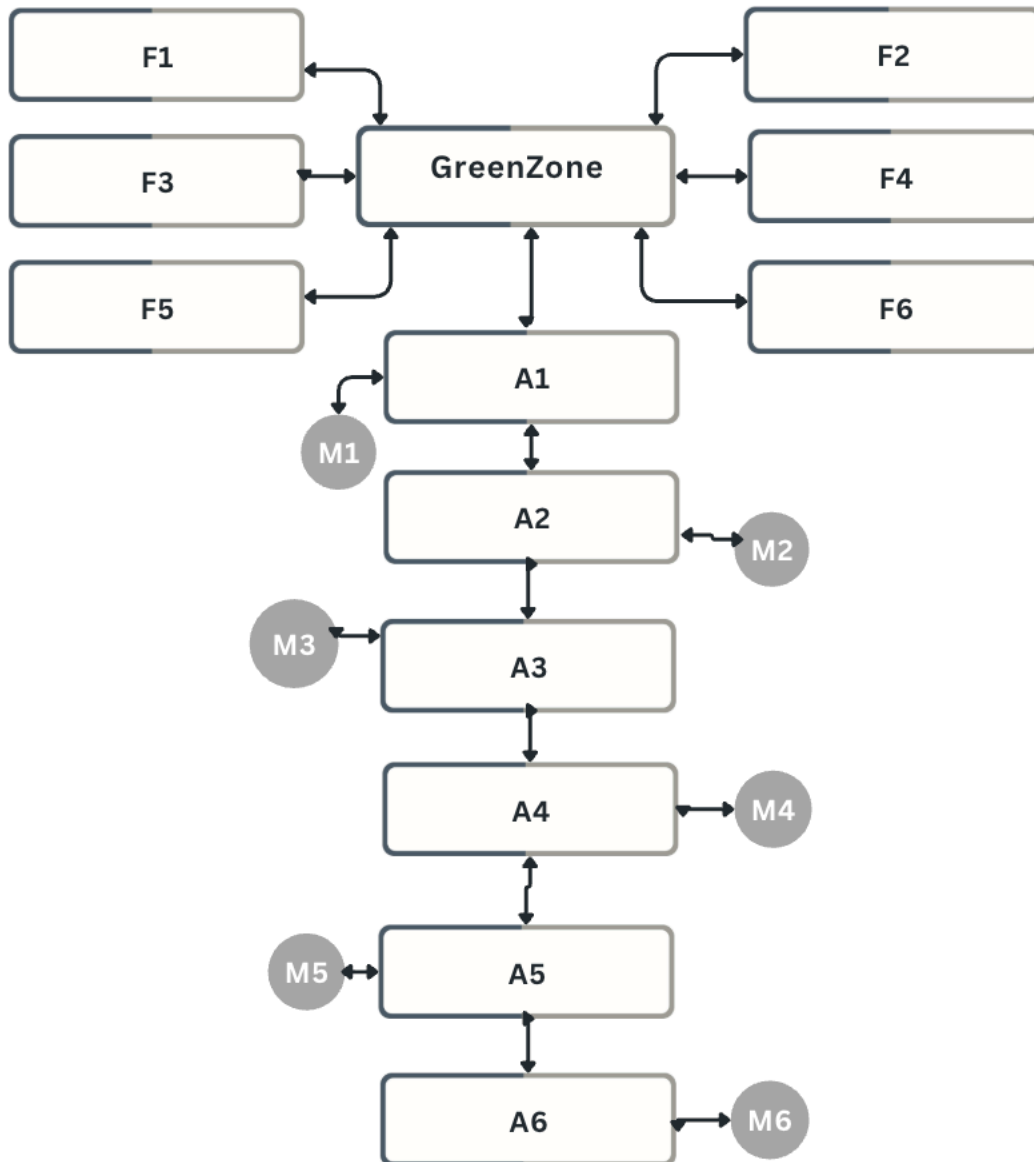


- **A5.4:** Consegna Versione 1.0 del codice sorgente
- **A6 (Revisione e rilascio finale):**
 - **A6.1:** Revisione completa della documentazione finale
 - **A6.2:** Allineamento tra codice e documenti
 - **A6.3:** Rilascio della versione definitiva 1.1

5.1.3 Milestone principali

- **M1:** Consegna Piano di Progetto
- **M2:** Consegna Documento dei Requisiti
- **M3:** Consegna Piano di Testing
- **M4:** Versione 1.0 pronta per consegna interna
- **M5:** Versione 1.0 del codice completata
- **M6:** Versione 1.1 del codice e allineamento documentazione

5.1.4 WBS



5.2 Dipendenze

Per **dipendenze** si intendono quelle attività che sono strettamente legate al completamento di una o più attività precedenti. Alcune fasi del progetto, infatti, non possono iniziare finché non sono stati completati i risultati o i documenti prodotti dalle fasi precedenti.



Le attività con dipendenze obbligate vengono quindi considerate **sequenziali**, mentre altre, indipendenti, possono procedere in parallelo.

Nella tabella seguente sono riportate le principali attività del progetto GreenZone, con la loro durata stimata e le date di inizio e fine più precoci (**ES, EF**) e più tardive (**LS, LF**).

Da tali informazioni è possibile derivare il **cammino critico**, ovvero la sequenza di attività fondamentali per il rispetto dei tempi di consegna.

Tutte le attività identificate (A1–A6) sono considerate parte del cammino critico, poiché ciascuna di esse rappresenta una fase indispensabile e sequenziale per il completamento del progetto. Un eventuale ritardo in una qualsiasi di queste attività comporterebbe il posticipo dell'intero piano temporale.

5.2.1 Tabella dipendenze

ID	Dipendenze	Durata	ES	EF	LS	LF
A1	-	7gg	07/10/2025	14/10/2025	07/10/2025	14/10/2025
A2	A1	4gg	14/10/2025	18/10/2025	14/10/2025	18/10/2025
A3	A2	22gg	18/10/2025	09/11/2025	18/10/2025	09/11/2025
A4	A3	6gg	09/11/2025	15/11/2025	09/11/2025	15/11/2025
A5	A4	37gg	15/11/2025	22/12/2025	15/11/2025	22/12/2025
A6	A5	35gg	22/12/2025	26/01/2026	13/12/2025	26/01/2026

Legenda:

- 1 – ES: Earliest start time
- 2 – EF: Earliest finish time
- 3 – LS: Latest start time
- 4 – LF: Latest finish time



5.3 Risorse necessarie

Per la realizzazione del progetto **GreenZone** si individuano diverse categorie di risorse fondamentali:

- **Risorse umane:** costituite dai membri del team, ognuno con competenze specifiche in analisi, sviluppo, progettazione grafica, testing e documentazione.
- **Risorse hardware:** comprendono i computer personali dei membri, utilizzati per lo sviluppo e la redazione dei documenti, e i dispositivi Android necessari per il test dell'applicazione.
- **Risorse software:** includono gli strumenti necessari allo sviluppo e alla gestione del progetto, come **Android Studio** per il coding, **GitHub** per il versionamento del codice, **Google Drive** per l'archiviazione condivisa della documentazione, e piattaforme di comunicazione come **Discord** e **WhatsApp** per il coordinamento del gruppo.
- **Risorse temporali:** rappresentano il tempo destinato alla pianificazione, allo sviluppo e al collaudo dell'applicazione, oltre a quello dedicato all'apprendimento degli strumenti tecnologici utilizzati. La corretta distribuzione delle ore di lavoro è essenziale per rispettare le scadenze dei deliverable.
- **Risorse documentative:** comprendono la documentazione di riferimento necessaria per lo sviluppo, come le guide ufficiali delle API Android, la documentazione di Firebase e le specifiche dei requisiti del progetto.

5.3.1 Diagramma di PERT



5.3.2 Stima dei costi

Il progetto **GreenZone** non prevede costi elevati, poiché la maggior parte delle risorse è già a disposizione dei membri del team o si basa su strumenti gratuiti.

I principali costi riguardano l'hardware, in particolare i computer personali utilizzati per lo sviluppo e uno o due telefoni Android destinati ai test dell'applicazione, per un totale di circa **600 euro**.

Per quanto riguarda il software, vengono utilizzati strumenti open source come **Android Studio**, **GitHub**, **Firebase** e **Google Drive**, che non comportano spese di licenza. Si possono aggiungere circa **50 euro** di costi generali (connessione Internet, energia elettrica e piccoli accessori) e **20–30 euro** per eventuali risorse di formazione o spazio cloud aggiuntivo.



In totale, il costo complessivo stimato per la realizzazione del progetto è di circa **700–750 euro**, una cifra sostenibile grazie all'utilizzo di strumenti gratuiti e alle risorse personali del gruppo.

5.4 Allocazione del Budget e delle risorse

5.4.1 Deadlines e Milestone

Il progetto **GreenZone** segue un calendario preciso scandito da sette deliverable principali, che rappresentano le tappe fondamentali per la gestione e il monitoraggio dello stato di avanzamento.

Ogni milestone coincide con la conclusione di una fase significativa e con la consegna del relativo documento o versione software.

- **07/10/2025 – D0**
Definizione del gruppo di lavoro e assegnazione dei ruoli principali.
- **14/10/2025 – D1**
Presentazione della proposta iniziale e approvazione dell'idea progettuale.
- **21/10/2025 – D2 / M1**
Consegna del Piano di Progetto e definizione dell'organizzazione interna.
- **11/11/2025 – D3 / M2**
Redazione e consegna del Documento dei Requisiti funzionali e non funzionali.
- **18/11/2025 – D4 / M3**
Conclusione del Piano di Testing e definizione delle procedure di verifica.
- **30/11/2025 – D5 / M4**
Completamento del Documento di Progettazione e delle specifiche architetture.
- **22/12/2025 – D6 / M5**
Rilascio della Versione 1.0 del codice sorgente e avvio della fase di verifica finale.
- **26/01/2026 – D7 / M6**
Consegna della Versione 1.1 definitiva e allineamento della documentazione di progetto.

5.4.2 Diagramma di Gantt

Il **Diagramma di Gantt** descrive graficamente la distribuzione temporale delle attività, mettendo in evidenza la sequenza cronologica e la durata di ogni fase.

Ogni attività (**A1–A6**) è associata al proprio deliverable e segue il cammino critico, che attraversa tutte le fasi del progetto.



Attività	Descrizione	Inizio	Fine	Durata (gg)															
A1	Analisi e setup del progetto	07/10/2025	14/10/2025	7															
A2	Architettura e mockup iniziale	14/10/2025	18/10/2025	4															
A3	Implementazione funzionalità principali	18/10/2025	09/11/2025	22															
A4	Integrazione FE/BE e test intermedio	09/11/2025	15/11/2025	6															
A5	Ottimizzazione, usabilità e versione 1.0	15/11/2025	22/12/2025	37															
A6	Revisione finale e versione 1.1	22/12/2025	26/01/2026	35															

5.4.3 Scheduling di Progetto

Lo scheduling prevede una distribuzione **sequenziale e continua** delle attività, secondo un approccio **Agile a sprint bisettimanali**.

Ogni sprint comprende pianificazione, sviluppo, testing e revisione dei risultati ottenuti.

- **Ottobre 2025:** Fase di avvio, analisi dei requisiti e definizione dell'architettura (A1–A2).
- **Novembre 2025:** Implementazione delle funzionalità principali e primo test intermedio (A3–A4).
- **Dicembre 2025:** Ottimizzazione, debugging e rilascio versione 1.0 (A5).
- **Gennaio 2026:** Revisione della documentazione, rifinitura e consegna della versione 1.1 finale (A6).

Lo scheduling rispetta perfettamente le milestone previste e permette un monitoraggio costante dell'avanzamento tramite incontri settimanali su **Discord** e **WhatsApp**, oltre al controllo delle versioni tramite **GitHub**.