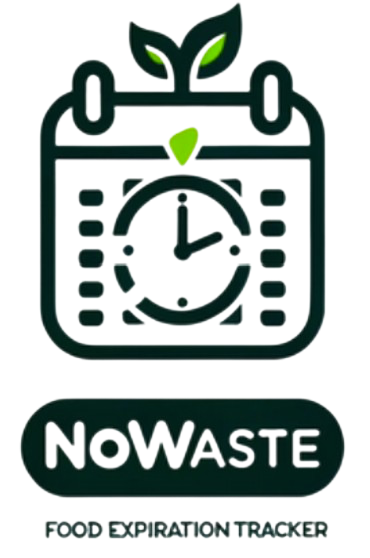
**Progetto di Ingegneria del Software 2023/24**

**Università Ca’ Foscari Venezia**

****

**Piano di Progetto**

2.0

**DDM**

29/01/24

*Document Informations*

|  |  |  |
| --- | --- | --- |
| NomeProgetto | Acronimo |  |
| Deliverable | Piano di Progetto |  |
| Data di Consegna | 29 / 01 / 24 |  |
| Team Leader | Martina Ragusa 885113@stud.unive.it |  |
| Team members | Daniela Spac 885386@stud.unive.it  Diego Marigo 853421@stud.unive.it |  |

*Document History*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Issue Date | Stage | Changes | Contributors |
| 1.0 | 03 / 10 / 23 | Draft | Creazione Piano di Progetto | MR, DS, DM |
| 1.1 | 16/10/2023 | Draft | Stesura Piano di Progetto | MR, DS, DM |
| 2.0 | 29/01/2024 | Final | \_ | MR, DS, DM |

**Indice**

[1.1 INTRODUZIONE 4](#_heading=h.gjdgxs)

[1.2.](#_heading=h.30j0zll) DELIVERABLES DEL PROGETTO 4

[1.3.](#_heading=h.1fob9te) EVOLUZIONE 5

[1.4.](#_heading=h.3znysh7) MATERIALE DI RIFERIMENTO 5

[1.5.](#_heading=h.2et92p0) DEFINIZIONI E ABBREVIAZIONI 5

[2.](#_heading=h.tyjcwt) ORGANIZZAZIONE DEL PROGETTO 6

[2.1.](#_heading=h.3dy6vkm) MODELLO DEL PROCESSO 6

[2.2.](#_heading=h.1t3h5sf) STRUTTURA ORGANIZZATIVA 6

[2.3.](#_heading=h.4d34og8) INTERFACCE ORGANIZZATIVE 6

[2.4.](#_heading=h.2s8eyo1) RESPONSABILITÀ DI PROGETTO 6

[3.](#_heading=h.17dp8vu) PROCESSI GESTIONALI 7

[3.1.](#_heading=h.3rdcrjn) GESTIONE DEI RISCHI 7

[3.2.](#_heading=h.26in1rg) MECCANISMI DI MONITORAGGIO E DI CONTROLLO 11

[3.3.](#_heading=h.lnxbz9) PIANIFICAZIONE DELLO STAFF 12

[4.](#_heading=h.35nkun2) PROCESSI TECNICI 12

[4.1.](#_heading=h.1ksv4uv) METODI, STRUMENTI, TECNICHE 12

[4.2.](#_heading=h.44sinio) DOCUMENTAZIONE DEL SOFTWARE 13

[4.3.](#_heading=h.2jxsxqh) FUNZIONALITÀ DI SUPPORTO AL PROGETTO 13

[5.](#_heading=h.z337ya) PIANIFICAZIONE DEL LAVORO, DELLE RISORSE UMANE E BUDGET 14

[5.1.](#_heading=h.3j2qqm3) WBS (WORK BREAKDOWN STRUCTURE) 14

[5.2.](#_heading=h.1y810tw) DIPENDENZE 14

[5.3.](#_heading=h.4i7ojhp) RISORSE NECESSARIE 15

[5.4.](#_heading=h.2xcytpi) ALLOCAZIONE DEL BUDGET E DELLE RISORSE 15

[5.5.](#_heading=h.1ci93xb) PIANIFICAZIONE 16

## 1.1 INTRODUZIONE

Il progetto in questione si focalizza sullo sviluppo di un'applicazione Android denominata NoWaste che assiste gli utenti nel monitorare le date di scadenza degli alimenti acquistati, al fine di ridurre lo spreco di cibo e promuovere un consumo più consapevole. L'applicazione permetterà di:

* Registrarsi fornendo alcuni dati personali, e definire le preferenze in termini di notifiche e altri metodi d’avviso;
* Aggiungere alimenti all'applicazione, specificando il nome, la data di scadenza e la quantità;
* Visualizzare un elenco degli alimenti inseriti, ordinati in base alla data di scadenza;
* Ricevere notifiche quando un alimento sta per scadere, con un anticipo personalizzabile dall'utente;
* Modificare o eliminare gli alimenti inseriti, o contrassegnare un alimento come consumato;
* Condividere una lista della spesa con le date di scadenza con altri utenti, facilitando la gestione condivisa di una dispensa o frigorifero in ambienti condivisi come appartamenti o uffici;
* Gestire le impostazioni dell'account e personalizzare le preferenze dell'applicazione, come la frequenza delle notifiche e la visualizzazione degli alimenti.

Il progetto mira a sensibilizzare gli utenti sul problema dello spreco alimentare e a fornire uno strumento semplice ed efficace per gestire e consumare gli alimenti in modo responsabile, promuovendo così un impatto positivo sia a livello individuale che collettivo.

## 1.2. DELIVERABLES DEL PROGETTO

|  |  |  |
| --- | --- | --- |
| **Nome Deliverable** | **Data Consegna** | **Luogo di Consegna** |
| Piano di Progetto | 17/10/2023 | Moodle del corso |
| Documento di 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 documentazione | 15/01/2024 | Moodle del corso |

## 

## 1.3. EVOLUZIONE

NoWaste è un'applicazione destinata a crescere e adattarsi alle esigenze dei nostri utenti, nonché a contribuire in modo significativo alla riduzione dello spreco alimentare.

Un ruolo fondamentale nello sviluppo l’ha giocato sicuramente il feedback degli utenti, che ci ha permesso di capire al meglio quali funzionalità possiedono la priorità affinché l’applicazione incontri le loro necessità.

## 1.4. MATERIALE DI RIFERIMENTO

* [Slides del corso di Ingegneria del Software](https://moodle.unive.it/course/view.php?id=11421)
* [Materiale del team](https://github.com/UniveIngSw2020/CaptureTheFlag) [*Junipero*](https://github.com/UniveIngSw2020/CaptureTheFlag)[per il corso di Ingegneria](https://github.com/UniveIngSw2020/CaptureTheFlag) [del Software dell’anno accademico 2020-21](https://github.com/UniveIngSw2020/CaptureTheFlag)
* Materiale del team *BreakingBad* per il corso di Ingegneria del Software dell’anno accademico 2022-23
* [StackOverflow per eventuali problematiche simili riscontrate da altri Sviluppatori](https://stackoverflow.com/)
* [Documentazione Google per Android](https://developer.android.com/reference/)
* [Android Studio](https://developer.android.com/studio)
* [FireBase per sviluppo di Basi di Dati](https://firebase.google.com/docs)

## 1.5. DEFINIZIONI E ABBREVIAZIONI

* [**Android**](https://it.wikipedia.org/wiki/Android)è un sistema operativo per dispositivi mobili sviluppato da Google, progettato principalmente per sistemi embedded quali smartphone e tablet;
* **App** In informatica, un'applicazione mobile è un'applicazione software dedicata ai dispositivi di tipo mobile, quali smartphone o tablet;
* [**Firebase**](https://it.wikipedia.org/wiki/Firebase)è una piattaforma per la creazione di applicazioni per dispositivi mobili e web sviluppata da Google.
* [**WhatsApp**](https://it.wikipedia.org/wiki/WhatsApp)è un'applicazione statunitense non libera di messaggistica istantanea centralizzata, creata nel 2009 da WhatsApp, Inc.;
* [**Google Drive**](https://it.wikipedia.org/wiki/Google_Drive)è un servizio web, in ambiente cloud computing, di memorizzazione e sincronizzazione online introdotto da Google;
* [**Documenti Google**](https://it.wikipedia.org/wiki/Documenti_Google)è un programma gratuito basato sul Web di elaborazione di testi;
* [**GitHub**](https://it.wikipedia.org/wiki/GitHub)è un servizio di hosting per progetti software, e permette tramite l’integrazione di git di tenere sotto controllo le diverse versioni del codice sorgente;
* [**Android Studio**](https://it.wikipedia.org/wiki/Android_Studio)è un ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaforma Android;
* [**Diagramma di Gantt**](https://it.wikipedia.org/wiki/Diagramma_di_Gantt)è uno strumento di supporto alla gestione dei progetti;
* [**PERT**](https://it.wikipedia.org/wiki/PERT/CPM)è un metodo statistico di determinazione dei tempi delle attività di progetto (ma può essere applicato anche ai costi);
* [**Kanban**](https://it.wikipedia.org/wiki/Kanban)in ambito di sviluppo software si fa riferimento a Kanban come una metodologia di sviluppo software.

# ORGANIZZAZIONE DEL PROGETTO

## MODELLO DEL PROCESSO

Il gruppo ha deciso di sviluppare questo progetto utilizzando il metodo a cascata. Questa decisione è stata influenzata dal desiderio di garantire una maggiore sicurezza nella scrittura del codice, ottenere una visione più chiara dell'intero processo di sviluppo del progetto e la flessibilità di apportare modifiche durante lo sviluppo senza il rischio di sprecare tempo e risorse preziose per soddisfare le esigenze degli utenti nel modo migliore possibile.

## STRUTTURA ORGANIZZATIVA

Il gruppo ha optato per un’organizzazione decentralizzata, in cui tutti i componenti si trovano allo stesso livello. In questo modo, le decisioni sono prese dall’intero team attraverso discussioni che permettono di vedere le eventuali problematiche da diversi punti di vista, trovando così soluzioni migliori. Inoltre, con questa struttura democratica, tutti i componenti sono sempre aggiornati sullo stato dell’intero progetto.

A blue sign with white text

Description automatically generated

## INTERFACCE ORGANIZZATIVE

Durante lo svolgimento ci sarà la necessità di comunicare con entità esterne al gruppo. Il team leader Martina Ragusa avrà il compito di informare periodicamente il Professor A. Cortesi sulla situazione del progetto oltre che alla consegna dei deliverables. Inoltre, tutti i membri del gruppo avranno il compito di seguire attentamente le lezioni e nel caso di dubbi porre domande sul tema organizzativo al Professor A. Cortesi e domande di tipo tecnico al Professor A. Spanò. Sarà necessario interfacciarsi con i futuri tester per avere feedback da loro e applicare modifiche in base ai problemi sorti.

## RESPONSABILITÀ DI PROGETTO

Dato che il gruppo è composto solamente da tre persone, non si effettuerà alcuna divisione in sottogruppi,

ma l’intero team lavorerà su tutti i fronti dello sviluppo.

# PROCESSI GESTIONALI

## GESTIONE DEI RISCHI

**3.1.1. Rischi Identificati**

1. **Errori nel codice**  
   - Categoria: Sviluppo  
   - Probabilità: 5 (molto alta)  
   - Impatto: Da 1 a 5  
   - Azione: Rivedere il codice e apportare modifiche.
2. **Difficoltà con il linguaggio di programmazione**   
   - Categoria: Sviluppo  
   - Probabilità: 2 (bassa)  
   - Impatto: 3   
   - Azione: Rivedere le regole di programmazione e documentarsi su Internet.
3. **Guasto hardware ai computer**  
   - Categoria: Materiale  
   - Probabilità: 1 (molto bassa)  
   - Impatto: 3 (salvo casi gravi quali rottura di tutti i pc)  
   - Azione: Procurarsi componenti di ricambio o, nel caso peggiore, una macchina che sostituisca quella guasta.
4. **Guasto software ai computer - Categoria: Sviluppo**  
   - Probabilità: 2 (bassa)  
   - Impatto: Da 3 a 5   
   - Azione: Cercare di sistemare l'errore documentandosi su Internet o, nel caso peggiore, formattare la macchina.
5. **Perdita del lavoro fatto**  
   - Categoria: Sviluppo  
   - Probabilità: 1 (molto bassa)  
   - Impatto: 5  
   - Azione: Ripristino via backup.
6. **Problemi di salute**  
   - Categoria: Personale  
   - Probabilità: 4 (alta)  
   - Impatto: Da 1 a 3 (salvo casi gravi)  
   - Azione: Riallocazione dei lavori per la settimana, nella speranza che la malattia sussida.
7. **Modifiche dei requisiti**  
   - Categoria: Progetto  
   - Probabilità: 3 (media)   
   - Impatto: Da 2 a 4   
   - Azione: Mantenere chiaro l'obiettivo dell'app per intercettare tutte le possibili sviste presenti nella progettazione, e ridurre al minimo le modifiche dei requisiti. Documentare eventuali modifiche.
8. **Problemi di comunicazione nel team**   
   - Categoria: Organizzativo  
   - Probabilità: 3 (media)  
   - Impatto: 4   
   - Azione: Organizzare riunioni regolari per discutere lo stato del progetto e chiarire gli obiettivi assegnati al singolo.
9. **Sforzo sottostimato**  
   - Categoria: Pianificazione  
   - Probabilità: 4 (alta)  
   - Impatto: Da 3 a 5  
   - Azione: Rivedere periodicamente il piano di progetto e adattarlo in base all'andamento effettivo del lavoro.

**3.1.2. Tabella dei rischi**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **RISCHI** | **CATEGORIA** | **PROBABILITÀ** | **IMPATTO** | **AZIONE** |
| 1 | Errori nel codice | Sviluppo | 5 | 1 - 5 | Rivedere il codice e apportare modifiche. |
| 2 | Difficoltà con il linguaggio | Sviluppo | 2 | 3 | Rivedere le regole di  programmazione e documentarsi su Internet. |
| 3 | Guasto hardware | Materiale | 1 | 3 (salvo casi gravi quali) | Procurarsi componenti di ricambio o, nel caso  peggiore, una macchina che sostituisca quella guasta. |
| 4 | Guasto software | Sviluppo | 2 | 3 – 5 | Cercare di sistemare l'errore documentandosi su internet  o, nel caso peggiore, formattare la macchina. |
| 5 | Perdita critica del lavoro fatto | Sviluppo | 1 | 5 | In caso di perdita irrecuperabile delle risorse, il progetto potrebbe dover essere chiuso. |
| 6 | Problemi di salute | Personale | 4 | 1 - 3 (salvo casi gravi) | Collaborare con gli altri membri e suddividersi il lavoro della persona ammalata a vicenda. |
| 7 | Modifiche dei requisiti | Progetto | 3 | 2-4 | Mantenere chiarezza sul bisogno che si desidera soddisfare con il progetto, per intercettare possibili sviste nella progettazione e documentare eventuali modifiche |
| 8 | Problemi di comunicazione nella squadra | Organizzativo | 3 | 4 | Organizzare riunioni regolari per favorire la comunicazione tra i membri della squadra e chiarire gli obiettivi assegnati. |
| 9 | Sforzo sottostimato | Pianificazione | 4 | 3-5 | Rivedere periodicamente il piano di progetto e adattarlo in base all’andamento effettivo del lavoro |
| 10 | Conoscenze tecniche insufficienti | Personale / Organizzativo | 4 | 4 | Individuare con chiarezza le lacune dei membri, per poterle colmare rapidamente |
| 11 | Errori non rilevati prima della consegna | Pianificazione | 3-4 | 3-4 | Identificare l’estensione dell’errore, ed inserirlo subito in un backlog apposito. |

Legenda probabilità dei rischi

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

**3.1.3. Prevenzione e monitoraggio dei rischi**

Per mitigare i rischi identificati durante lo sviluppo del progetto, adotteremo le seguenti misure preventive e strategie di monitoraggio:

1. **Errori nel codice**

* Prevenzione: uso di debugger, unit test per garantire il rispetto della specifica stabilita. Faremo forte uso di strumenti di analisi statica come Lint (lo standard di Android Studio), FindBugs (specifico per Java) e Detekt (specifico per Kotlin). Faremo il possibile per seguire le migliori pratiche di programmazione
* Monitoraggio: Effettuare regolarmente il controllo di errori nel codice scritto.

1. **Difficoltà con il linguaggio di programmazione**

* Prevenzione: Rivedere le regole di programmazione e documentarsi su Internet in caso di difficoltà.
* Monitoraggio: Verificare il codice scritto per garantire la conformità alle regole di programmazione.

1. **Guasto hardware**

* Prevenzione: Evitare di esporre i dispositivi a situazioni di pericolo.
* Monitoraggio: Effettuare controlli periodici dell'integrità dell'hardware.

1. **Guasto software**

* Prevenzione: Prestare attenzione a come e cosa si installa sul proprio dispositivo.
* Monitoraggio: Controllare eventuali segnali software anomali e intervenire tempestivamente.

1. **Perdita del lavoro fatto**

* Prevenzione: Salvare il lavoro su più posti diversi, inclusi sistemi cloud.
* Monitoraggio: Verificare che il lavoro sia stato sincronizzato online in modo regolare.

1. **Problemi di salute**

* Prevenzione: Limitare il fisico a situazioni di contagio.
* Monitoraggio: Controllare eventuali sintomi e collaborare con gli altri membri per suddividere il lavoro in caso di malattia.

1. **Modifiche dei requisiti**

* Prevenzione: Mantenere una comunicazione continua con il committente per intercettare possibili sviste nella progettazione e documentare eventuali modifiche.
* Monitoraggio: Verificare che le modifiche siano coerenti con i requisiti iniziali e documentarle adeguatamente.

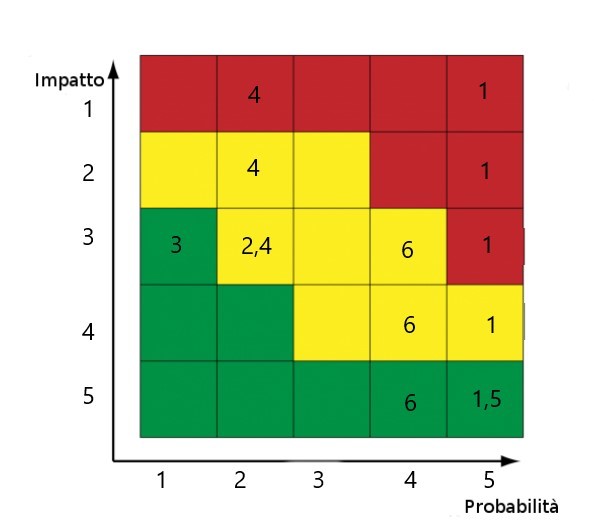
1. **Problemi di comunicazione nel team**

* Prevenzione: Organizzare riunioni regolari per favorire la comunicazione tra i membri del team e chiarire gli obiettivi assegnati.
* Monitoraggio: Monitorare la partecipazione e l'efficacia delle riunioni.

1. **Sforzo sottostimato**

* Prevenzione: Rivedere periodicamente il piano di progetto e adattarlo in base all'andamento effettivo del lavoro.
* Monitoraggio: Verificare periodicamente lo stato di avanzamento del lavoro e apportare aggiornamenti al piano se necessario.

**Modello dei rischi**



## MECCANISMI DI MONITORAGGIO E DI CONTROLLO

Il progetto verrà sviluppato in gruppo, di conseguenza il tutto sarà sotto l’occhio vigile dei componenti del team in modo da avere un controllo incrociato su documentazione e codice sorgente. Durante questo periodo ci saranno delle assegnazioni di diversi tasks per velocizzare determinate procedure di semplice implementazione, così facendo ci saranno controlli più intensi su porzioni di codice minori, migliorando sensibilmente il tutto. Per potersi confrontare si utilizzeranno i canali Discord e WhatsApp per tenere in comunicazione i membri del gruppo, Google Documents per la revisione della documentazione e GitHub per versionare il codice sorgente.

## PIANIFICAZIONE DELLO STAFF

Allo scopo di realizzare il progetto, i membri del team necessitano conoscenze del linguaggio Java, dei DBMS e intraprendere familiarità con l’ambiente di sviluppo Android. Allo scopo di quest'ultimo, ci si rifà alle lezioni del professor Spanò e all'approfondimento individuale.

Per il coordinamento dei lavori, l'idea iniziale è di usare gli strumenti di base forniti da Git. Semplici Issue per definire le task basilari ed i responsabili, e Milestone per gruppi di task che vanno completati per una scadenza. Si userà la pagina Project di Github per monitorare il progresso del progetto.

# PROCESSI TECNICI

## METODI, STRUMENTI, TECNICHE

#### Calcolatori

Lo sviluppo di tale progetto richiede l’utilizzo di un software per lo sviluppo di applicazioni Android. Per tale necessità si è pensato all’utilizzo di Android Studio che è disponibile per qualsiasi sistema operativo desktop e che permette la virtualizzazione di un dispositivo Android per il testing dell’applicazione, il che permette di semplificare lo sviluppo anche ai membri del gruppo che non possiedono un cellulare con sistema operativo Android installato.

Si è comunque ritenuto opportuno dover fare dei test anche su dispositivi fisici per evitare problematiche intrinseche dovute alla differenza tra dispositivi virtuali e fisici, oltre a problematiche dovute alla differenza delle versioni del sistema operativo; a tale necessità si è optato per l’utilizzo dei nostri dispositivi fisici.

#### Metodi di sviluppo

Come metodo di sviluppo abbiamo deciso di suddividere il progetto in diversi task. Tuttavia, dato che il gruppo è composto solamente da tre persone, non riteniamo necessario il dover assegnare task specifici a ciascuno. Cercheremo invece di lavorare tutti insieme su tutti i task (con delle possibili eccezioni) così da poterci confrontare e avere l’aiuto di tutti i componenti.

#### Tecniche

Abbiamo utilizzato Java come linguaggio di programmazione e la documentazione ufficiale di Android per approfondire, oltre ad aver sfruttato gli insegnamenti e i suggerimenti che ci ha dato il Professore Alvise Spanò riguardo l’utilizzo delle API Android.

## DOCUMENTAZIONE DEL SOFTWARE

Il software verrà documentato durante lo sviluppo del progetto, nello specifico verranno redatti i seguenti documenti:

* Piano di Progetto
* Documento di analisi e specifica
* Piano di testing
* Documento di Progettazione

I documenti nella loro versione finale verranno consegnati entro il 15/01/2024.

## FUNZIONALITÀ DI SUPPORTO AL PROGETTO

Sviluppare un progetto con codice di qualità è il principale obiettivo del gruppo.

In particolare, si desidera:

1. Tenere sempre aggiornata la documentazione (sia del codice che del progetto stesso);
2. Sfruttare i Design Patterns in modo da sviluppare un’applicazione con codice riutilizzabile e di facile comprensione;
3. Stendere codice efficiente in termini di tempo di esecuzione e di risorse utilizzate, visto che l’applicazione dovrà essere eseguita su dispositivi con una capacità di calcolo inferiore a dei calcolatori fissi o portatili;
4. Controllare ciclicamente il codice e testare l’applicazione in modo da rilevare possibili bugs o problematiche che potrebbero incidere sull’esperienza d’uso dell’utente.

Per lo sviluppo di tale progetto, il gruppo ha deciso di usufruire dei seguenti strumenti:

* **GitHub**: servizio che permette di salvare nel cloud il codice dell’applicazione, di sincronizzare il lavoro di tutti i membri del progetto e di tenere traccia di tutte le modifiche che vengono apportate, tenendo traccia anche della cronologia delle modifiche.
* **Google Drive**: servizio cloud di Google che permette a tutto il gruppo di condividere una cartella in cui sono salvati tutti i documenti e le fonti ottenute da ricerche online.
* **Documenti Google**: sono stati scelti di documenti e fogli Google per stendere la documentazione del progetto in quanto permettono di condividere la sessione di scrittura in modo da poter scrivere simultaneamente sullo stesso foglio.
* **Android Studio**: [Android Studio è un ambiente di sviluppo](https://www.bing.com/ck/a?!&&p=7ab0f9f88bde4505JmltdHM9MTY2NTc5MjAwMCZpZ3VpZD0yNzYyNzEwNS0zOGFjLTYwMGMtMWJmOC03ZWVmMzljNzYxODcmaW5zaWQ9NTU5Mg&ptn=3&hsh=3&fclid=27627105-38ac-600c-1bf8-7eef39c76187&psq=android+studio&u=a1aHR0cHM6Ly9pdC53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9TdHVkaW8&ntb=1) [integrato per lo sviluppo per la piattaforma Android](https://www.bing.com/ck/a?!&&p=7ab0f9f88bde4505JmltdHM9MTY2NTc5MjAwMCZpZ3VpZD0yNzYyNzEwNS0zOGFjLTYwMGMtMWJmOC03ZWVmMzljNzYxODcmaW5zaWQ9NTU5Mg&ptn=3&hsh=3&fclid=27627105-38ac-600c-1bf8-7eef39c76187&psq=android+studio&u=a1aHR0cHM6Ly9pdC53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9TdHVkaW8&ntb=1), che supporta i linguaggi di programmazione Java e Kotlin.
* **WhatsApp**: piattaforma di messaggistica istantanea usufruibile sia per dispositivi mobili (cellulari e tablet) che desktop.

# PIANIFICAZIONE DEL LAVORO, DELLE RISORSE UMANE E BUDGET

## WBS (WORK BREAKDOWN STRUCTURE)

Illustriamo qui la struttura del progetto nelle diverse funzioni, attività e milestones, ovvero la Work Breakdown Structure (WBS) del progetto.

Queste sono le funzioni identificate dal team:

* **Project Management**

Questo è il ruolo principale del project manager: esso dovrà coordinare le attività del team, pianificare incontri, risolvere questioni interne, monitorare il progresso del progetto e, in caso di problemi, attuare soluzioni, garantendo un'adeguata e tempestiva consegna di documenti e codice.

* **Training**

Implica un impegno costante, da parte del team, nell'acquisire conoscenze sulle tecnologie e i metodi necessari per lo sviluppo del progetto. Il team dovrà continuamente aggiornarsi e formarsi sui linguaggi di programmazione, sui framework e sulle pratiche di gestione dei progetti.

* **Documentazione**

La stesura dei documenti viene eseguita secondo le linee guida e le scadenze fornite dal Professore.

Tali documenti da presentare sono:

* Piano di progetto;
* Documento di analisi e specifica;
* Piano di testing;
* Documento di progettazione.
* **Quality control**

Funzione di controllo costante della qualità del servizio tramite confronti con gli utenti finali e con gli obiettivi di progetto.

Le due milestones identificate dal gruppo sono:

* **Versione Completa 1.0**: verranno ampliate le funzionalità di base affinché sia possibile l'interazione tra gli utenti tramite una sezione legata al singolo evento per discuterlo, fornendo più informazioni agli utenti.
* **Versione Finale 1.1**: verranno raccolti suggerimenti forniti dai tester, cercando di implementare i loro suggerimenti e risolvere eventuali bugs.

## DIPENDENZE

Le dipendenze sono tutte quelle attività che sono strettamente legate al termine di un’attività precedente. Alcune attività sono infatti dipendenti e hanno bisogno di alcune parti del progetto terminate, senza contare inoltre che la forza lavoro è limitata e può saturarsi facilmente. È quindi necessario distribuire il lavoro in maniera equa e dilatata nel tempo. Le attività che prevedono una dipendenza sono quindi forzate ad essere sequenziali, mentre le attività che non hanno obblighi l’una verso l’altra, è possibile parallelizzarle per ottimizzare i tempi di lavoro e permettere ai membri del team di lavorare separatamente su più attività contemporaneamente.

È stato inoltre definito un Diagramma di Gantt per rappresentare graficamente lo scheduling del progetto mostrando la suddivisione del lavoro in task (che abbiamo fatto coincidere con le consegne del progetto), evidenziandone l’ordine e stimando un tempo massimo di compimento per ciascuno di essi.

Una suddivisione più approfondita in tasks minori viene mostrata nel diagramma di Gantt presente alla [sezione 5.5](#_heading=h.1ci93xb).

#### Diagramma di Pert

Proposta iniziale

inizio 26/09

fine 03/10

durata 7 gg

Piano di progetto

inizio 03/10

fine 17/10

durata 14 gg

Documento dei requisiti

inizio 17/10

fine 31/10

durata 14 gg

Piano di testing

inizio 31/10

fine 14/11

durata 15 gg

Documento di progettazione

inizio 14/11

fine 28/11

durata 14 gg

Versione 1.0 codice sorgente

inizio 28/11

fine 15/12

durata 17 gg

Versione 1.1 + documentazione

inizio 15/12

fine 15/01

durata 31 gg

## RISORSE NECESSARIE

Le risorse necessarie alla realizzazione del progetto includono:

* Risorse umane: i 3 componenti del gruppo e le loro competenze tecniche.
* Risorse hardware: dispositivi hardware a disposizione del team utilizzati in tutte le fasi di sviluppo e testing.
* Risorse software: comprendono tutti i software che saranno utilizzati per la buona riuscita del progetto, per esempio: Android Studio, Google Drive, GitHub e piattaforme di comunicazione che permettono al team di rimanere sempre in contatto (WhatsApp, GoogleMeet).
* Risorse temporali: corrispondono alle ore lavorative necessarie allo sviluppo e alla buona realizzazione del progetto.

## ALLOCAZIONE DEL BUDGET E DELLE RISORSE

Per la realizzazione del progetto gli strumenti software utilizzati sono gratuiti (Android Studio, Google Drive, Github…).

Per quanto riguarda i costi dei lavoratori, supponiamo che ciascun membro del team lavorerà per un totale di 12 ore a settimana, di cui 8 ore nel fine settimana e il resto nei giorni feriali. Dato che il team ha iniziato a lavorare il 19 settembre 2023 e la scadenza del progetto è il 15 gennaio 2024, abbiamo un totale di 20 settimane di lavoro, senza considerare i giorni festivi. Su questa base, il totale delle ore lavorative del gruppo ammonta a circa 1200 ore. Tenendo conto di uno stipendio medio in Italia, pari a 13 euro all'ora, il preventivo complessivo si attesta a 15.600 euro.

## PIANIFICAZIONE

Il progetto seguirà la pianificazione stabilita dal professor Cortesi nel corso di Ingegneria del Software (Anno Accademico 2023/2024):

1. Proposta iniziale - 03/10/2023
2. Piano di Progetto - 17/10/2023
3. Documento dei requisiti - 31/10/2023
4. Piano di testing - 14/11/2023
5. Documento di Progettazione - 28/11/2023
6. Versione 1.0 del codice sorgente (Milestone) - 15/12/2023
7. Versione 1.1 del codice sorgente e allineamento documentazione (Milestone) - 15/01/2024

Come primo passo, dopo aver individuato il tema del progetto e aver steso il piano di progetto, il gruppo ha iniziato a condividere idee per migliorare ed ampliare l’idea principale. Abbiamo proseguito elencando tutti i requisiti per la migliore implementazione dell’immagine che avevamo in mente dell’applicazione, sulla base dei quali abbiamo steso il documento relativo. Abbiamo poi iniziato la stesura del codice dell’app, partendo dalle sue schermate di fondazione. Tra queste, individuiamo le schermate di login e registrazione, le quali permettono all’utente di accedere all’app, proseguendo poi con la schermata per la visualizzazione delle liste di alimenti dell’utente, la quale costituisce la parte principale del progetto.

Sulla base di ciò che stavamo sviluppando, abbiamo poi individuato i test necessari che dovevano essere eseguiti per garantire il miglior utilizzo dell'applicazione da parte degli utenti. Successivamente, sono state ultimate le interfacce utente (in termini di design), grazie alle quali abbiamo poi potuto proseguire allo sviluppo di tutte le parti dell’applicazione.

In parallelo con lo sviluppo dell’applicazione, oltre ad individuare ulteriori test e requisiti necessari per perfezionare ulteriormente l’esperienza utente, sono state apportate le modifiche alla documentazione, affinché fosse sempre aggiornata.

Possiamo avere una visione di tutte le attività nel diagramma di Gantt sottostante.

