

## Sommario

<b>Introduzione.....</b>	<b>1</b>
<b>Software Life Cycle.....</b>	<b>1</b>
<b>Caratteristiche del processo XP adottate: .....</b>	<b>1</b>
<b>Organizzazione degli sprint: .....</b>	<b>1</b>
<b>Configuration Management.....</b>	<b>2</b>
<b>Documentazione e Strumenti Utilizzati .....</b>	<b>2</b>
<b>Struttura del Repository GitHub .....</b>	<b>3</b>
<b>People Management and Team Organization.....</b>	<b>3</b>

## Introduzione

Questo documento definisce come verrà gestito il software UnibgPark facendo capire come il team avrà intenzione di operare:

## Software Life Cycle

Il processo di sviluppo adottato è basato sulla metodologia **Extreme Programming (XP)**, che è un metodo puramente agile in cui si programma la maggior parte delle volte in coppia se non in trio in modo da avere sempre un pilota e uno o più co-pilota che lo aiutano. La scelta di XP si basa sull'importanza della flessibilità, della collaborazione e della qualità del prodotto finale.

Caratteristiche del processo XP adottate:

- **Planning Game:** Riunioni settimanali per definire funzionalità e obiettivi.
- **Incontri frequenti:** Incontri dal vivo per scambiarsi idee e per programmare insieme contemporaneamente
- **Small Releases:** Rilasci frequenti per ricevere feedback continuo.
- **Pair Programming:** Programmazione in coppia per garantire una revisione costante.
- **Collective Ownership:** Il codice è accessibile a tutti i membri del team.
- **Continuous Integration:** Integrazione frequente per identificare problemi rapidamente.

Organizzazione degli sprint:

Il team ha organizzato il lavoro in timebox che variavano da una o due settimane. Con un gruppo composto da tre membri, la comunicazione è stata rapida ed efficace, permettendo una revisione regolare dei processi e delle attività su base giornaliera. Si programmava a coppie, dove ci si concentrava su un obiettivo specifico e di apportare un contributo diretto e collaborativo al progresso del progetto.

- **(25/11/2024 - 01/12/2024):** Progettazione del database, inclusa la definizione delle tabelle per utenti, parcheggi e pagamenti, e l'integrazione iniziale dei sensori con un prototipo di backend.

- **(02/12/2024 - 15/12/2024):** Sviluppo dell'app mobile per visitatori occasionali, comprensiva di interfaccia per il pagamento orario, gestione dei QRCode e notifiche push.
- **(16/12/2024 - 07/01/2025):** Implementazione delle funzionalità di gestione abbonamenti per residenti, come la richiesta e il rinnovo di abbonamenti tramite sistema web o app.
- **(08/01/2025 - 17/01/2025):** Testing finale, integrazione completa del sistema (sensori, database e app), correzione di eventuali bug e rilascio della prima versione funzionante.

## Configuration Management

Le attività del progetto vengono gestite attraverso issue dettagliate, monitorate e organizzate tramite una **Kanban Board** integrata su GitHub. Questo approccio consente una visione sempre aggiornata dello stato di avanzamento del progetto, semplificando il processo di sviluppo e favorendo l'integrazione continua delle modifiche. La creazione e l'assegnazione degli issue ai membri del team permettono di definire compiti specifici e assicurare una gestione sistematica delle attività. Inoltre, l'utilizzo di **milestones** facilita il monitoraggio del completamento delle task e degli obiettivi principali.

Per la gestione della configurazione, il progetto utilizza **GitHub**, supportato da **GitHub Desktop** per il controllo dei repository locali e l'esecuzione di operazioni come pull, commit e push. Questa combinazione garantisce un controllo efficace delle versioni e una collaborazione fluida tra i membri del team.

Eventuali problematiche o errori rilevati durante lo sviluppo vengono segnalati tempestivamente tramite l'apertura di issue. L'issue viene poi assegnata al membro del team più adatto a gestirlo, favorendo una risoluzione rapida e mirata. Ogni nuova funzionalità o correzione di bug è sviluppata in branch dedicati, per garantire un flusso di lavoro organizzato e minimizzare i rischi di conflitti.

Le modifiche al codice vengono sottoposte a revisione tramite **pull request**, che assicurano un controllo di qualità prima dell'integrazione nel branch principale. Questo processo strutturato, unito a una comunicazione efficace attraverso GitHub, supporta un ciclo di sviluppo ordinato e altamente collaborativo.

### Uso di Issues, Branches, Pull Requests e Code Reviews

- Ogni attività è stata gestita tramite **issue dettagliate**, che descrivono le funzionalità o le correzioni da implementare.
- Le modifiche sono state sviluppate su **branch separati**, garantendo un isolamento delle funzionalità e una maggiore sicurezza.
- Al completamento di ogni attività, è stata creata una **Pull Request (PR)**, sottoposta a revisione da un altro membro del team, per assicurare coerenza e qualità del codice.
- Le Pull Request non approvate sono state aggiornate con le correzioni necessarie prima del merge nel branch principale.

### Documentazione e Strumenti Utilizzati

- **Papyrus UML:** Per creare e gestire i diagrammi UML.
- **Eclipse:** per la programmazione del codice
- **GitHub:** Per la visione delle nuove versioni e la gestione del lavoro collaborativo.
- **Google Meet:** Per riunioni e aggiornamenti del team.

## Struttura del Repository GitHub

Il repository del progetto è organizzato in sezioni per garantire una gestione ordinata dei componenti e un accesso semplice e una gestione ottimale dei vari componenti. Le principali sezioni saranno le seguenti:

- **Codice:** Vi saranno tutti i file per la visione del programma
- **Diagrammi UML:** Questa sezione raccoglierà i modelli, come diagrammi delle classi, delle sequenze e delle attività, utilizzati per rappresentare in modo chiaro e visivo la struttura e le funzionalità del software.
- **Documentazione:** Includerà materiali relativi alla pianificazione del progetto, i vari problemi riscontrati, all'ingegneria dei requisiti, le attività di manutenzione e molto altro
- **Images:** Conterrà file grafici, ad esempio il logo dell'attività
- **Test:** Ospiterà script di test e report dettagliati per verificare il corretto funzionamento e le prestazioni del sistema.

Questa struttura organizzativa garantirà una gestione efficiente e una collaborazione agevolata all'interno del team.

## People Management and Team Organization

Il lavoro del team si è basato sulle pratiche di **Extreme Programming (XP)**, che promuovono una stretta collaborazione, ruoli intercambiabili e un ritmo di lavoro sostenibile.

### Organizzazione del Team

Il team è composto da tre membri che condividono responsabilità simili, alternando i ruoli a seconda delle necessità del progetto.

### Modalità di Lavoro

- **Riunioni settimanali:** Pianificazione degli sprint e revisione del lavoro completato, cercando di vedersi di persona in modo da poi programmare a quattro mani
- **Pair Programming:** Per migliorare la qualità del codice e favorire la condivisione delle conoscenze.
- **Test-Oriented Development (TDD):** Creazione dei test prima dello sviluppo delle funzionalità.