



# **Progetto di Ingegneria del Software 2022/23**

## **Università Ca' Foscari Venezia**

### **Piano di Progetto**

1.2

**Team Angio**

13/01/2023



### Document Informations

| Progetto Standard | MyLocalBooking                                   | MLB  |
|-------------------|--|--|
| Deliverable       | Piano di progetto                                |  |
| Data di Consegna  | 16/10/2022                                       |  |
| Team Leader       | Pietro Visconti                                  | <a href="mailto:885448@stud.unive.it">885448@stud.unive.it</a>   |
| Team members      | Nicola Marizza<br>Pietro Donega<br>Mirco Mellara | <a href="mailto:887004@stud.unive.it">887004@stud.unive.it</a><br><a href="mailto:881909@stud.unive.it">881909@stud.unive.it</a><br><a href="mailto:882963@stud.unive.it">882963@stud.unive.it</a> |

### Document History

| Version | Issue Date     | Stage     | Changes                     | Contributors                      |
|---------|----------------|-----------|-----------------------------|-----------------------------------|
| 1.0     | 16 / 10 / 2022 | Draft     | Creazione documento         | Tutti                             |
| 1.1     | 18 / 10 / 2022 | Revisione | Rimozione membro del gruppo | Mellara Mirco                     |
| 1.2     | 13 / 01 / 2023 | Finale    | Finalizzazione documento    | Visconti Pietro,<br>Mellara Mirco |



# Piano di progetto

## Indice

|  |           |
|--|-----------|
| <b>Indice</b>  | <b>1</b>  |
| <b>Intoduzione</b>   | <b>3</b>  |
| <b>Overview</b>  | <b>3</b>  |
| Deliverables   | 4         |
| Evoluzione   | 4         |
| Materiali  | 4         |
| Definizioni e abbreviazioni.                                       | 4         |
| <b>Organizzazione del progetto</b>                                 | <b>5</b>  |
| Modello del processo   | 5         |
| Struttura organizzativa  | 5         |
| Interfacce organizzative   | 5         |
| Responsabilità di Progetto   | 6         |
| <b>Processi Gestionali</b>   | <b>7</b>  |
| Obiettivi e Priorità   | 7         |
| Assunzioni, Dipendenze, Vincoli                                    | 7         |
| Gestione dei rischi  | 7         |
| Meccanismi di monitoraggio e di controllo                          | 9         |
| Pianificazione dello staff   | 10        |
| <b>Processi tecnici</b>  | <b>10</b> |
| Metodi, Strumenti e Tecniche                                       | 10        |
| Documentazione del Software  | 10        |
| Funzionalità di supporto al progetto                               | 10        |
| <b>Pianificazione del lavoro, delle risorse umane e del budget</b> | <b>11</b> |
| WBS (Work Breakdown Structure)                                     | 11        |
| Dipendenze   | 12        |
| Risorse necessarie   | 12        |
| Allocazione del budget e risorse                                   | 12        |
| Pianificazione   | 13        |
| <b>Manuale gestione rischi</b>                                     | <b>15</b> |



# Introduzione

## Overview

### Target di utenza

L'utilizzo di questa applicazione è destinato ad un pubblico tipicamente formato da giovani o adulti che desiderano usufruire di servizi sotto forma di spazi prenotabili nella loro zona di residenza o in una zona specifica prefissata. In particolare gli utenti con età inferiore di 18 anni necessitano della supervisione dei loro genitori.

### Tipologie di utenti

- 1) Gli utenti che usufruiscono dei servizi / spazi offerti;
- 2) Gli utenti (in qualità di organizzazioni o persone individuali) che offrono servizi / spazi.

### Obiettivo principale

Questa applicazione permette ai suoi utenti, in base alla categoria di appartenenza, di mettere a disposizione o prenotare degli spazi privati.

L'idea che sta dietro a questo progetto è la semplificazione di accesso e di comunicazione tra organizzazioni o privati che mettono a disposizione i propri spazi e la popolazione dei vari quartieri. L'obiettivo finale è quello di rendere facile e veloce le prenotazioni di servizi nel proprio quartiere, semplificando il processo di comunicazione tra le due parti.

## Deliverables

Tutti i materiali che verranno consegnati, saranno forniti al docente via e-mail da parte del capogruppo.

D0: Proposta iniziale (03/10/2022)

D1: Piano di Progetto (17/10/2022)

D2: Documento dei Requisiti (31/10/2022)

D3: Piano di Testing (14/11/2022)

D4: Documento di Progettazione (28/11/2022)

D5: Versione 1.0 del codice sorgente (15/12/2022)

D6: Versione 1.1. del codice e allineamento documentazione (15/01/2023)



## Evoluzione

### Possibili cambiamenti / funzionalità aggiuntive da implementare

- Possibilità di pagare i servizi all'interno dell'app tramite servizio esterno.
- Possibilità di cambiare tema dell'app (tema scuro, tema chiaro).
- Possibilità di esportare le prenotazioni su google calendar.
- Inviare segnalazioni agli sviluppatori.
- Sistema di notifiche in app.
- Annunci intorno a sé.
- Blacklist di utenti per provider con relative operazioni di blacklisting e un-blacklisting e striking
- Rimuovere un establishment per i Provider

## Materiali

- Slide del corso
- Videolezioni su Android Studio del professor Spanò
- Documentazione di Android Studio
- Documentazione Ruby on Rails
- Documentazione PostgreSQL
- Documentazione Java

## Definizioni e abbreviazioni.

**Team:** un gruppo di persone dedicate ad un determinato aspetto dello sviluppo dell'applicazione (i.e. API)

**API:** Application Programming Interface, modulo del software esposto all'ambiente esterno il cui compito è quello di ricevere richieste e restituire una risposta.

**DB:** Database



# Organizzazione del progetto

## Modello del processo

Viene utilizzato il modello a **cascata**: le attività sono suddivise in cicli di sviluppo, in cui ogni ciclo è strettamente dipendente dal precedente. Al termine di ogni settimana lavorativa viene organizzata una riunione collettiva in cui si decide come distribuire il lavoro dei task successivi.

## Struttura organizzativa

Abbiamo scelto di operare tramite una struttura organizzativa “**Per progetto**”.

In particolare abbiamo deciso di suddividere lo sviluppo in due “sotto-progetti” a cui lavorano due “Team”, che si occupano dello sviluppo del “Front-end” e del “Back-end” relativi all’applicazione.

La struttura che gestisce l’organizzazione del personale per cui abbiamo optato è quella “**Democratica decentralizzata**”, in cui tutti i membri del gruppo di lavoro decidono insieme come risolvere i problemi e organizzare il lavoro.

Anche se abbiamo designato delle figure di “Capogruppo” e “Project Manager”, tali posizioni non offrono poteri decisionali aggiuntivi, ma servono più per aiutare nella coordinazione del lavoro di gruppo, specialmente nei casi in cui vi siano difficoltà nel trovare punti d’incontro.

## Interfacce organizzative

- Discord
- Github
- WhatsApp
- Google Docs
- Google Sheets
- Potenziali utenti tramite interviste



## Responsabilità di Progetto

|                                      |                 |
|--------------------------------------|-----------------|
| Capogruppo                           | Visconti Pietro |
| Project manager                      | Mellara Mirco   |
| Function manager (Documentazione)    | Mellara Mirco   |
| Function manager (Controllo qualità) | Marizza Nicola  |
| Responsabile consegne                | Visconti Pietro |
| Responsabile interviste/osservazione | Mellara Mirco   |
| Responsabile progettazione front-end | Mellara Mirco   |
| Responsabile realizzazione front-end | Donega Pietro   |
| Responsabile database                | Marizza Nicola  |
| Responsabile API                     | Visconti Pietro |
| Responsabile Testing                 | Mellara Mirco   |



# Processi Gestionali

## Obiettivi e Priorità

Nella nostra organizzazione del progetto puntiamo a suddividere il lavoro in due “Team”, uno responsabile dell’aspetto “back-end” e uno dell’aspetto “front-end” dell’applicazione, al fine di lasciare ai membri una singola “area” su cui focalizzarsi, senza dover richiedere ai componenti una conoscenza vasta e approfondita di diverse “aree” di sviluppo.

Le priorità dello sviluppo sono mantenere un rapporto quanto più orizzontale possibile, dove le decisioni vengono prese quanto più spesso possibile dall’insieme del gruppo o quanto meno del “Team”, concordando tutto a priori durante le riunioni settimanali.

## Assunzioni, Dipendenze, Vincoli

- Fornire numero di telefono, nome, cognome (o nome organizzazione)
- Fornire data di nascita
- Chi offre uno spazio deve esserne il proprietario
- L’app necessita connessione di rete
- E’ consigliato l’accesso alla posizione
- La posizione dell’utente influisce pesantemente sulla quantità e la tipologia di offerte disponibili nella sua area





## Gestione dei rischi

**Legenda:** RU - Risorse Umane;

RR - Requisiti;

RT - Tecnologici

**Scala impatto:** 1 - Risolvibile in giornata → 5 - Possibile fallimento del progetto

| Rischio  | ID | Categoria | Impatto | Probabilità | RMMM  |
|--|----|-----------|---------|-------------|---|
| Ritiro di uno dei membri del gruppo.   | 1  | RU        | 4       | 10%         | <a href="#">Caso 3</a>                          |
| Il team non ha intesa.   | 2  | RU        | 4       | 25%         | <a href="#">Caso 4</a>                          |
| Possibile malattia.  | 3  | RU        | 1       | 75%         | <a href="#">Caso 1</a> o <a href="#">Caso 2</a> |
| Possibile errore umano nel sistema di versioning.                              | 4  | RU        | 3       | 85%         | <a href="#">Caso 4</a>                          |
| Quantità di funzionalità troppo pretenziose rispetto ai tempi.                 | 5  | RR        | 3       | 30%         | <a href="#">Caso 3</a>                          |
| Possibile interruzione / malfunzionamento dei servizi di hosting del database. | 6  | RT        | 1       | 10%         | <a href="#">Caso 1</a>                          |
| Carenza di conoscenze tecniche di sviluppo Android.                            | 7  | RT        | 2       | 40%         | <a href="#">Caso 1</a>                          |
| Sovrapposizionamento di altri corsi di studi.                                  | 8  | RU        | 4       | 40%         | <a href="#">Caso 4</a>                          |
| Cambiamento di esigenze software.  | 9  | RT        | 5       | 15%         | <a href="#">Caso 4</a>                          |
| Cambiamento date scadenze  | 10 | RR        | 3       | 5%          | <a href="#">Caso 3</a>                          |
| Requisiti mal definiti   | 11 | RR        | 3       | 50%         | <a href="#">Caso 1</a>                          |



|   |    |    |   |     |   |
|---|----|----|---|-----|---|
| Piano di progetto non definito sufficientemente | 12 | RR | 3 | 20% | <a href="#">Caso 1</a>                          |
| Piano di testing non abbastanza esaustivo       | 13 | RR | 2 | 20% | <a href="#">Caso 1</a>                          |
| Sviluppo di funzionalità non previste           | 14 | RR | 4 | 20% | <a href="#">Caso 2</a> o <a href="#">Caso 3</a> |



miro

## Meccanismi di monitoraggio e di controllo

Il presupposto principale di questa attività di controllo consiste nella possibilità di osservare e misurare regolarmente la produttività del progetto.

Come meccanismi di monitoraggio e controllo includiamo:

- Misurazione dell'avanzamento delle attività del progetto (dove ci troviamo in persona e discutiamo a voce di come sta procedendo il progetto), tramite le milestones



- Confronto con le previsioni del piano di progetto che costituiscono la baseline del progetto (dove dovremmo essere)
- Sorveglianza periodica (indicativamente ogni 2-3 giorni) da parte dei responsabili per il corretto avanzamento della progettazione (sono stati assegnati dei responsabili per ogni tipo di attività).

## Pianificazione dello staff

- ★ Pietro V: proposto di sviluppare l'API tramite Ruby on Rails per via dell'esperienza maturata con il linguaggio Ruby e il framework stesso. Si occupa inoltre dello sviluppo lato backend della applicazione attraverso le richieste alla API. Invece il ruolo di capogruppo è stato scelto in maniera democratica.
- ★ Nicola: si propone di gestire tutti gli aspetti legati al database, in quanto ha una buona dimestichezza con l'RDBMS PostgreSQL avendo diversi progetti personali alle spalle, alcuni dei quali ancora attivi. Si occupa inoltre dello sviluppo lato backend della applicazione attraverso le richieste alla API. Inoltre si propone di gestire gli aspetti legati alla qualità del software.
- ★ Mirco: offerto di lavorare alla documentazione per via di esperienze passate, dove ha maturato dimestichezza con i vari strumenti utili a tale funzione quali Google Docs e Google Sheets. Il ruolo di "Project manager" invece, è stato deciso dal gruppo. Si occupa inoltre della progettazione e realizzazione delle interfacce grafiche.
- ★ Pietro D: proposto di lavorare sulla realizzazione grafica.

## Processi tecnici

### Metodi, Strumenti e Tecniche

- Android Studio come IDE di sviluppo Android
- Linguaggio Java, Ruby, SQL
- Carta e penna per le bozze dei design
- PgAdmin e DBeaver per lo sviluppo del DB.
- Render.com per l'hosting del DB



## Documentazione del Software

Il modello di documentazione del software consiste nella consegna della versione 1.0 prima delle scadenze prefissate e con la conseguente modifica/finalizzazione prima della consegna finale.

## Funzionalità di supporto al progetto

Verranno eseguiti controlli per garantire gli aspetti dello standard ISO/IEC 25010 più importanti per il progetto.

I vari controlli si suddivideranno in 3 categorie:

1. controlli periodici
2. controlli di fine ciclo di sviluppo
3. controlli finali

### Controlli periodici

Verranno svolti periodicamente lungo l'intera durata del progetto e saranno accessori alle altre categorie di controlli.

Sono incentrati prevalentemente su aspetti di qualità interna del software e sono i seguenti:

1. Verifica della qualità del codice (chiarezza, sinteticità, modularità, SOLID principles) di modo tale da minimizzare l'impatto di eventuali cambiamenti di requisiti quali aggiunta, rimozione o modifica di feature. Questa verifica è cruciale per mantenere un buon livello di manutenibilità del software, in quanto assicura la modularità, analizzabilità, testabilità, e modificabilità del software.
2. Assicurare una costante produzione di documentazione di qualità, in modo che ciascun team alla fine di ogni ciclo di sviluppo possa allinearsi facilmente con il lavoro degli altri team.
3. Verifica di una corretta gestione dello strumento di versioning, in particolare assicurando una ben definita compartimentalizzazione del lavoro dei diversi team allo scopo di prevenire conflitti e favorire un meccanismo semplice e intuitivo di merging dei branch.
4. Verifica del rispetto dei requisiti di sicurezza, in particolare definendo in modo netto le diverse classi di utenti e riservando ad ognuna le funzionalità del software più sensibili in base alle loro aree di competenza. Questa suddivisione sarà definita lato client, lato server e lato DBMS e contribuirà a garantire aspetti importanti della sicurezza quali riservatezza, integrità, non ripudio, responsabilità, autenticità.



## Controlli di fine ciclo di sviluppo

Questi controlli saranno implicitamente affiancati ai controlli periodici sopra descritti.

Ci sarà inoltre una particolare attenzione volta a tutti gli aspetti di idoneità funzionale (completezza, correttezza, adeguatezza) del ciclo di sviluppo appena conclusosi in virtù del ciclo di sviluppo successivo. La completezza e l'adeguatezza saranno verificate tramite l'analisi del codice, la correttezza tramite opportuno testing.

## Controlli finali

Saranno comunque affiancati dai controlli periodici e dai controlli di fine ciclo di sviluppo.

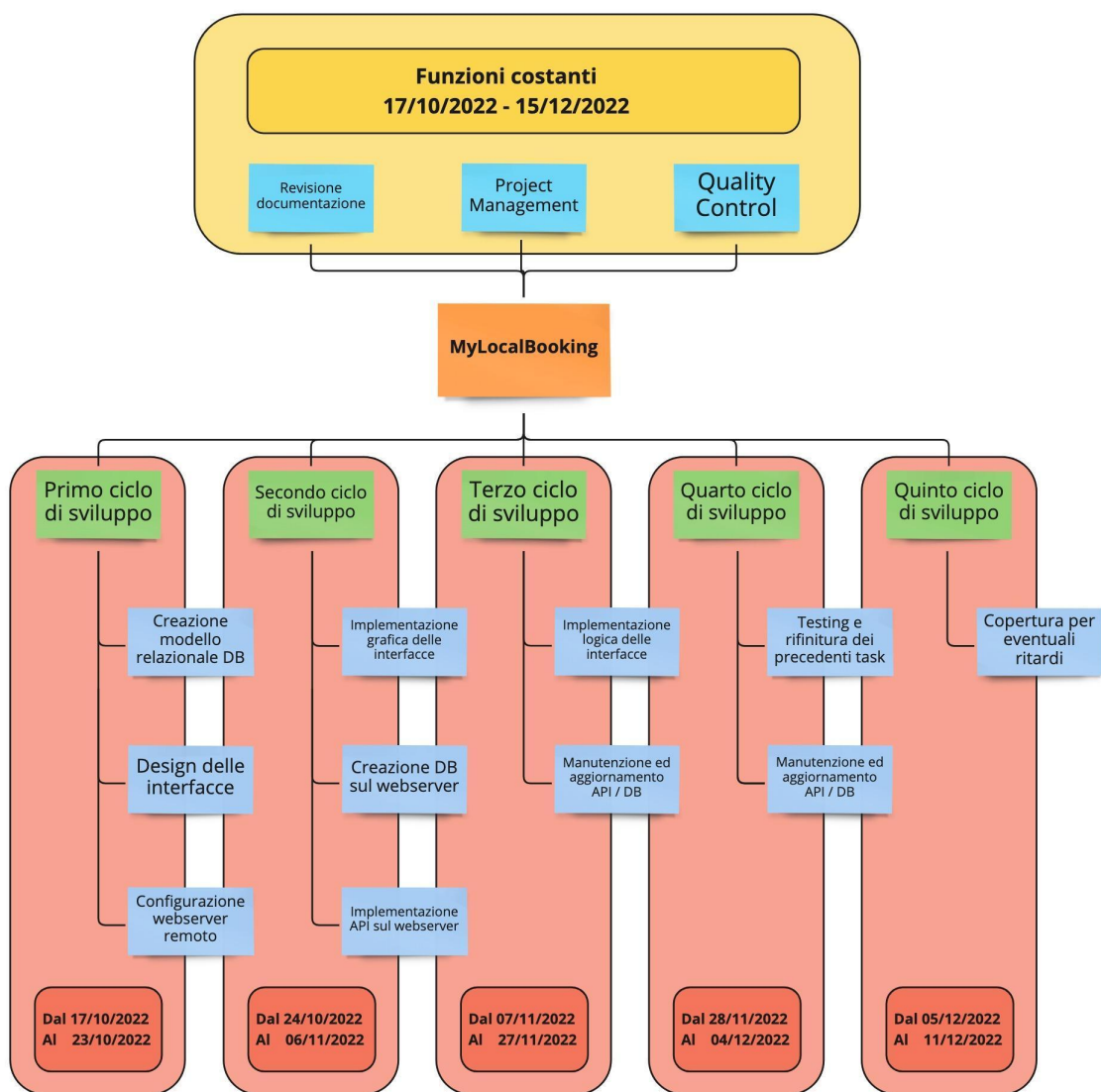
Consisteranno in test incentrati sugli aspetti della qualità esterna e della qualità in uso, con particolare attenzione verso gli aspetti di operabilità, apprendibilità e protezione degli errori.

Saranno eseguiti in due fasi della produzione del software: al termine del quarto ed eventualmente del quinto ciclo di sviluppo. Ogni aspetto che dovesse eventualmente essere considerato insoddisfacente durante i controlli del quarto ciclo sarà approfondito e migliorato durante il quinto.



# Pianificazione del lavoro, delle risorse umane e del budget

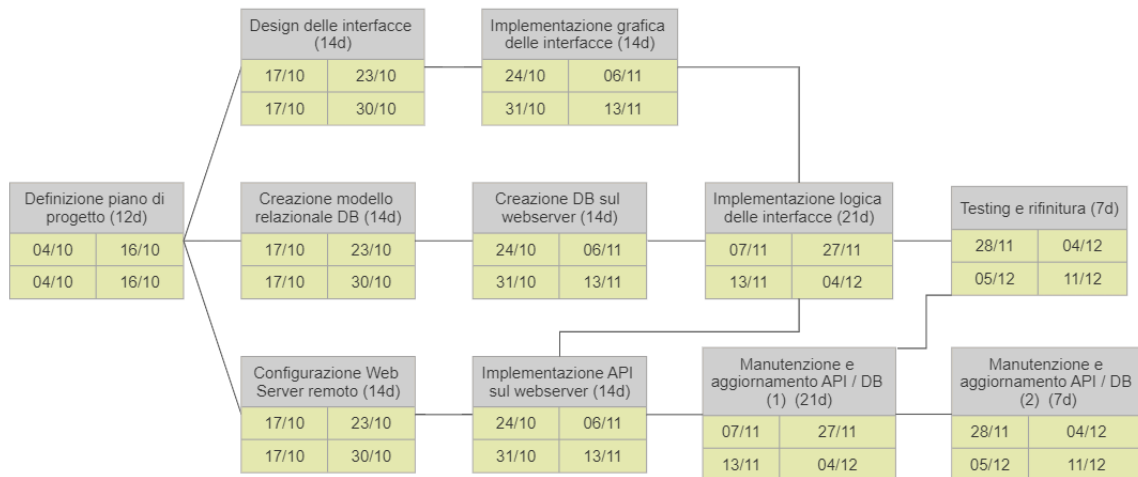
## WBS (Work Breakdown Structure)



miro



## Dipendenze



## Risorse necessarie

Abbiamo previsto di impiegare tutti i 4 membri del gruppo per pensare e realizzare il Software per tutta la durata prevista.

Tale durata è stimata essere 11 settimane lavorative, con partenza il giorno 26/09/2022 e terminazione prevista il giorno 11/12/2022.

Questa durata si riferisce alla versione 1.0 del Software, che con alte probabilità si differenzierà da quella finale di consegna che verrà presentata il giorno 15/01/2023. Le risorse necessarie dopo la consegna della versione 1.0, purtroppo non sono stimabili a priori e quindi aggiorneremo questa parte del documento nel momento in cui si rivelerà opportuno.

A livello di personale, non prevediamo l'aggiunta di nuovi membri nel gruppo, nemmeno per situazioni in cui membri abbandonino il progetto.

Mentre per risorse materiali, quali Hardware particolari, non prevediamo alcuna spesa aggiuntiva, visto che ogni membro è dotato a priori del necessario per svolgere le proprie mansioni.

## Allocazione del budget e risorse

Il costo temporale di ogni attività si può trovare nella WBS. A livello di budget invece, fortunatamente non prevediamo spese effettive di alcun tipo.

Ponendo l'ipotesi che, ogni membro del gruppo sia un programmatore "entry-level", in Italia lo stipendio medio lordo percepito da tale tipologia di lavoratori si aggira intorno agli €12,40 ([Fonte](#)).



Il calcolo effettuato prevede che €25.000 vengano distribuiti in 12 mensilità, per un totale di circa €2083,33 al mese. Assumendo che il lavoro consista di 168 ore mensili, ossia 40 settimanali, si ricava appunto la cifra di circa €12.40 lordi all'ora.

Partendo da questo punto, possiamo provare a stimare il costo del progetto, facendo altre assunzioni, tra cui:

- Ogni membro del gruppo percepisce lo stesso stipendio, a prescindere dalle responsabilità.
- Si stima (a priori) una media di **20** ore settimanali impiegate da ciascun membro per lo sviluppo di questa applicazione
- Non vengono considerate altre spese, come ad esempio di materiale hardware o licenze di software particolari.
- La data 26/09 è da considerarsi la prima giornata di lavoro, mentre la data 11/12 è considerata l'ultima giornata di lavoro effettivo.

Si nota quindi che la durata è di 11 settimane. Per cui si stima un costo settimanale secondo la formula:  $\text{costo\_medio\_orario} \times \text{numero\_membri} \times \text{ore\_settimanali\_a\_persona}$  che corrisponde a:  
 $€12,40 \times 5 \times 20 = \mathbf{€1240,00}$

Quindi per 11 settimane un costo complessivo di  $€1240,00 \times 11 = \mathbf{€13.640,00}$

**Nota:** La stima del monte ore fatta a priori con molta probabilità differirà dal monte ore effettivo di lavoro effettuato, per tanto alla consegna finale del progetto, verrà effettuata una seconda stima dei costi, basata sul lavoro effettivo.

Inoltre la stima effettuata non prende in considerazione aggiustamenti o modifiche successive alla consegna della versione 1.0 in data 15/11/2022 che è da reputarsi situazione poco realistica.

### **Aggiornamento:**

Avendo tenuto traccia del lavoro fatto da ciascun membro del gruppo e di quante ore abbia dedicato ad ogni task tramite un foglio "Documenti Google", possiamo fornire ora una stima più accurata del costo.

Mediamente i 4 membri del gruppo (perché vi è stato l'abbandono/espulsione di uno dei 5 originali), hanno dedicato a testa circa 130 ore.

Il costo totale si aggirerebbe quindi a  $€12,40 \times 4 \times 130 = \mathbf{€7.200}$  circa, visto anche che non sono serviti investimenti ulteriori





## Pianificazione

| Attività                                 | Scadenza   | Milestone   | Status       |
|--|------------|---|--------------|
| Creazione modello relazionale DB         | 23/10/2022 | Modello relazionale creato correttamente  | Completato ▾ |
| Design delle interfacce                  | 23/10/2022 | Scelti gli schizzi/bozze di interfacce che costituiranno il "front end" dell'applicazione                         | Completato ▾ |
| Configurazione Web Server remoto         | 23/10/2022 | Il servizio di render.com è attivo e utilizzabile   | Completato ▾ |
| Implementazione grafica delle interfacce | 06/11/2022 | Implementazione statica delle interfacce scelte all'interno dell'app.   | Completato ▾ |
| Creazione DB sul webserver               | 06/11/2022 | Il database è creato e hostato dal servizio in base al modello progettato   | Completato ▾ |
| Implementazione API sul webserver        | 06/11/2022 | L'API è correttamente eseguita e funzionante sul servizio di <i>render.com</i>                                    | Completato ▾ |
| Implementazione logica delle interfacce  | 27/11/2022 | Rendere le interfacce dinamiche e customizzabili di utente in utente  | Completato ▾ |
| Manutenzione ed aggiornamento API/DB (1) | 27/11/2022 | Tutti i servizi funzionano correttamente in base alle esigenze di sviluppo  | Completato ▾ |
| Testing e rifinitura precedenti task     | 04/12/2022 | Il processo di testing ha prodotto risultati soddisfacenti e tutti i task precedenti sono stati portati a termine | Completato ▾ |
| Manutenzione ed aggiornamento API/DB (2) | 04/12/2022 | Ottimizzazione dei servizi dell'API e versione finale   | Completato ▾ |



# Manuale gestione rischi

## Caso 1

Si effettua una revisione della suddivisione del lavoro per al massimo le due giornate successive, dove altri membri dello stesso “team” coprono i compiti del membro afflitto dal problema.

Questo tipo di rischio non dovrebbe comportare ritardi sulle tempistiche di progettazione, ma un momentaneo sovraccarico di lavoro per alcuni membri.

## Caso 2

Si effettua una revisione sostanziale della suddivisione del lavoro, che può prevedere anche lo spostamento di membri fra i vari “team”.

Questo tipo di rischio può impattare le tempistiche in modo più ostico, provocando anche svariati giorni di ritardo nella progettazione del Software, senza però causare un sovraccarico eccessivo nella mole di lavoro per ciascun membro.

## Caso 3

Si effettua una revisione totale della suddivisione del lavoro, che comporterà anche un aumento importante di lavoro per ogni membro.

Questo tipo di rischi provocherà un ritardo certo nelle tempistiche, potenzialmente anche di settimane, causando anche un considerevole aumento del carico di lavoro per ciascun membro.

## Caso 4

I membri del gruppo sospendono momentaneamente i task su cui stanno lavorando e tentano di rimediare al problema il prima possibile, tutti insieme.

Questo tipo di rischio provoca un ritardo non stimabile a priori, che dipenderà dalla velocità di risoluzione del problema. Nel caso migliore potrebbe essere una questione di poche ore, nel caso peggiore potrebbe portare a giornate intere di ritardo.