



# **/\* Suddivisione dei ruoli \*/**

**Ruben**

// Tester

**Francesco**

// Software Architect e assistente Programmatore



**Ivan**

/\* Programmatore e  
assistente  
Software Architect  
\*/

**Alessio**

// SCRUM Master

**Mirco**

// Presentatore Grafico del progetto

# /\* Linguaggio di programmazione e Brainstorming \*/

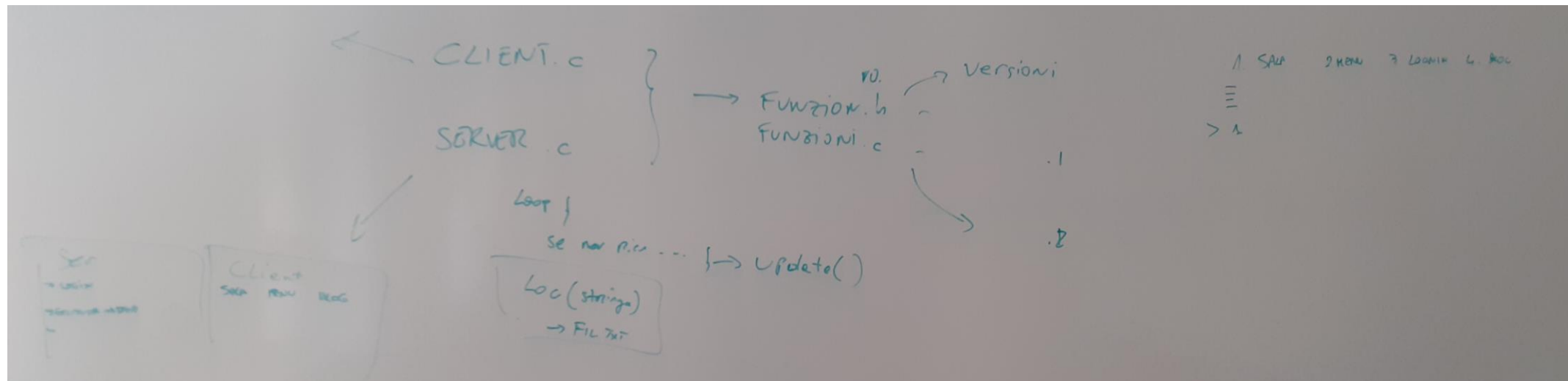
Come linguaggio di programmazione, è stato scelto **C** in quanto conosciuto da tutti i membri del gruppo.

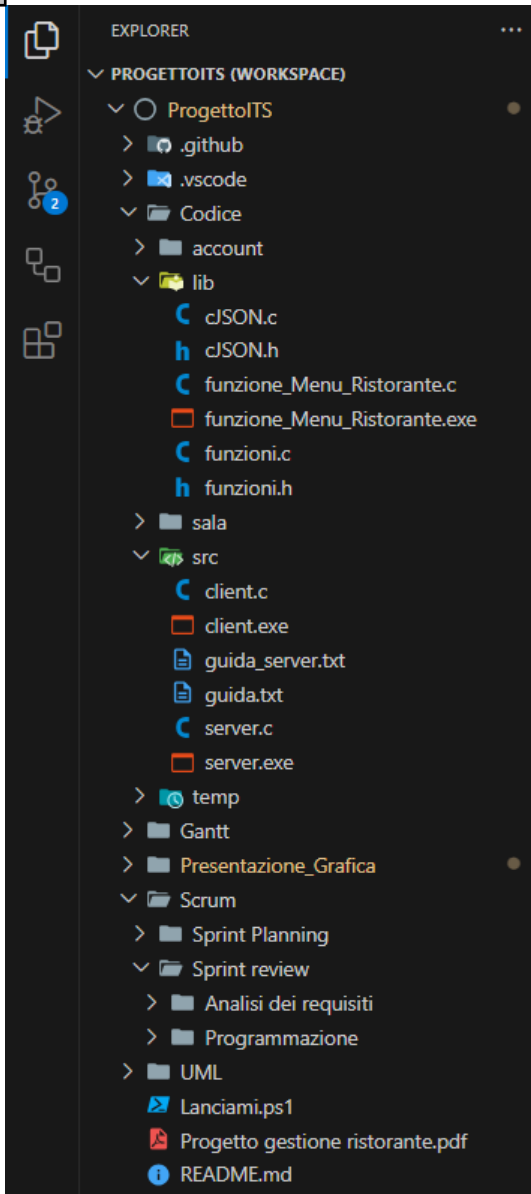
L'architetto del software ha suggerito un'**architettura Client / Server**, che interagiscono tra di loro tramite file temporanei.



```
#include <stdio.h>

void main()
{
    printf("La Gang del Bosco");
}
```





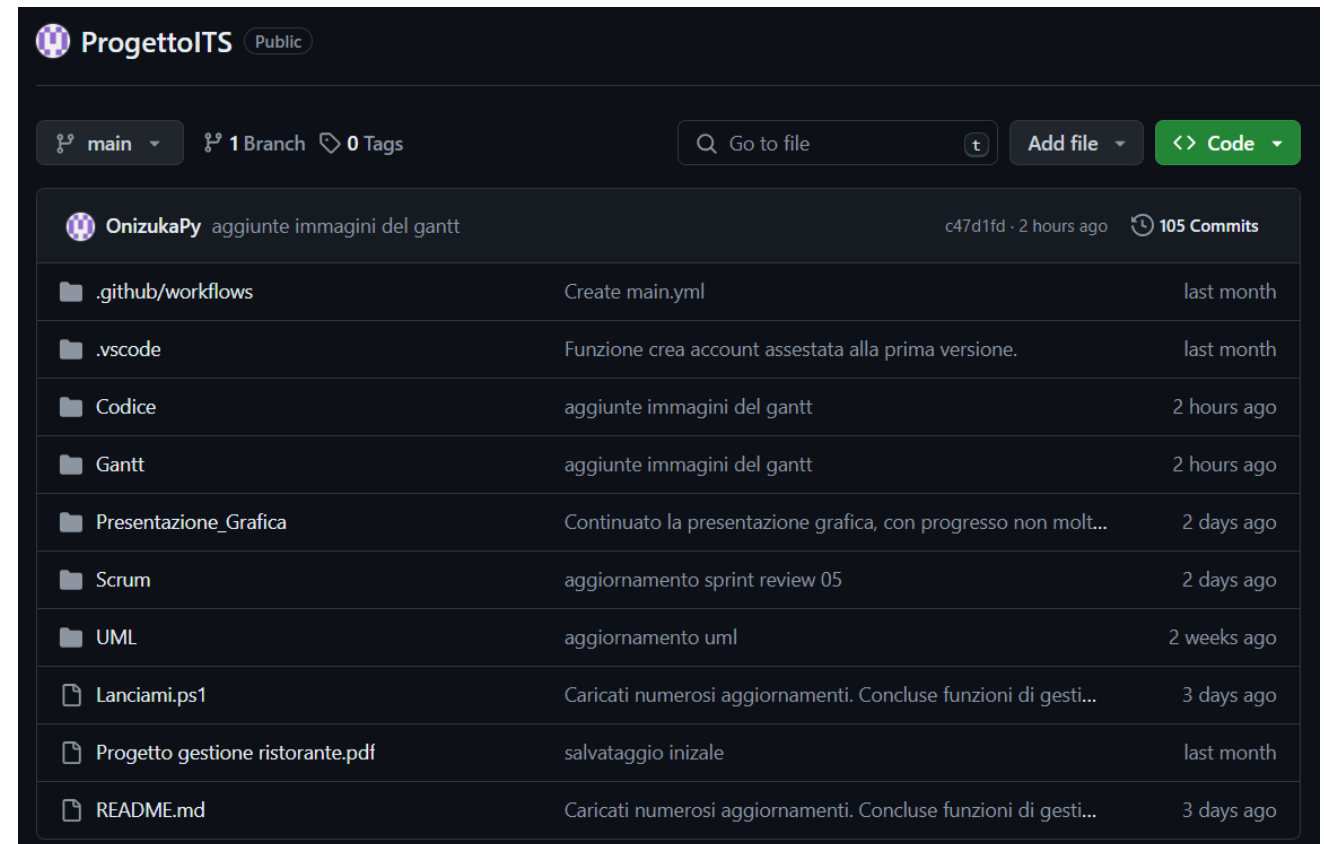
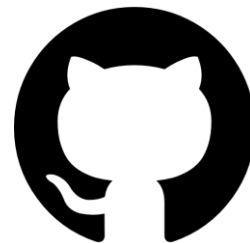
## /\* Ambiente di lavoro \*/

Per poter lavorare, il gruppo ha effettuato la sincronizzazione degli strumenti, installando **Visual Studio Code** come ambiente di sviluppo e collegandovi **GitHub** per committare gli aggiornamenti.

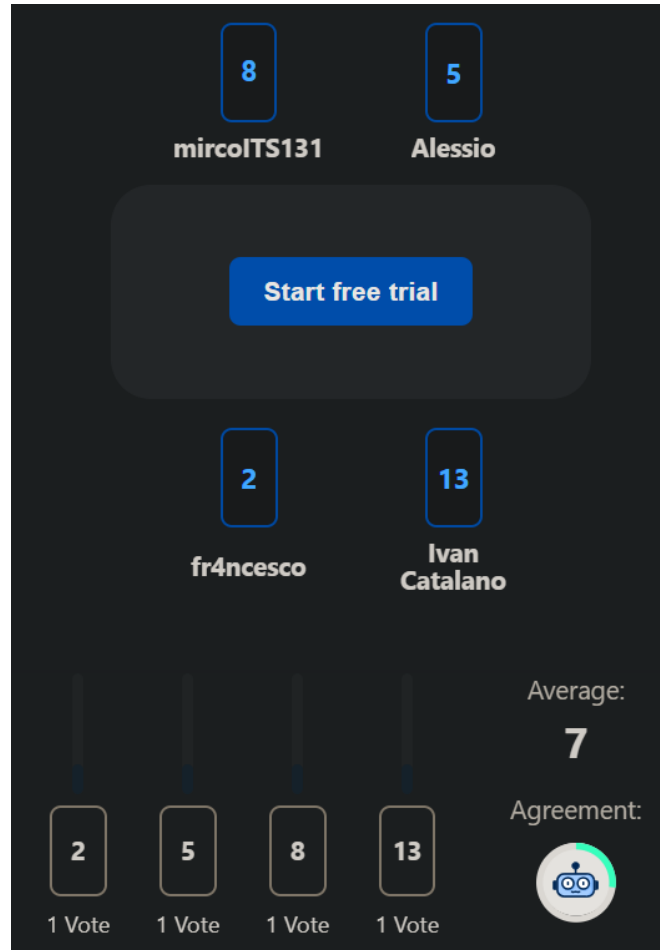


Visual  
Studio  
Code

GitHub



## /\* Analisi dei Requisiti \*/



## // SCRUM: Planning Poker

Valore attribuito alla realizzazione di una funzione di difficoltà media = 1

### Backlog

Calcolo delle stime delle singole task

Realizzazione dei diagrammi UML

Gestione utenti

Gestione dei Menù

Disposizione sala

Prenotazione dei tavoli

Gestione Ordini

Feedback clienti

### Punteggio (ore di lavoro)

1

3,7

4,3

4

3,3

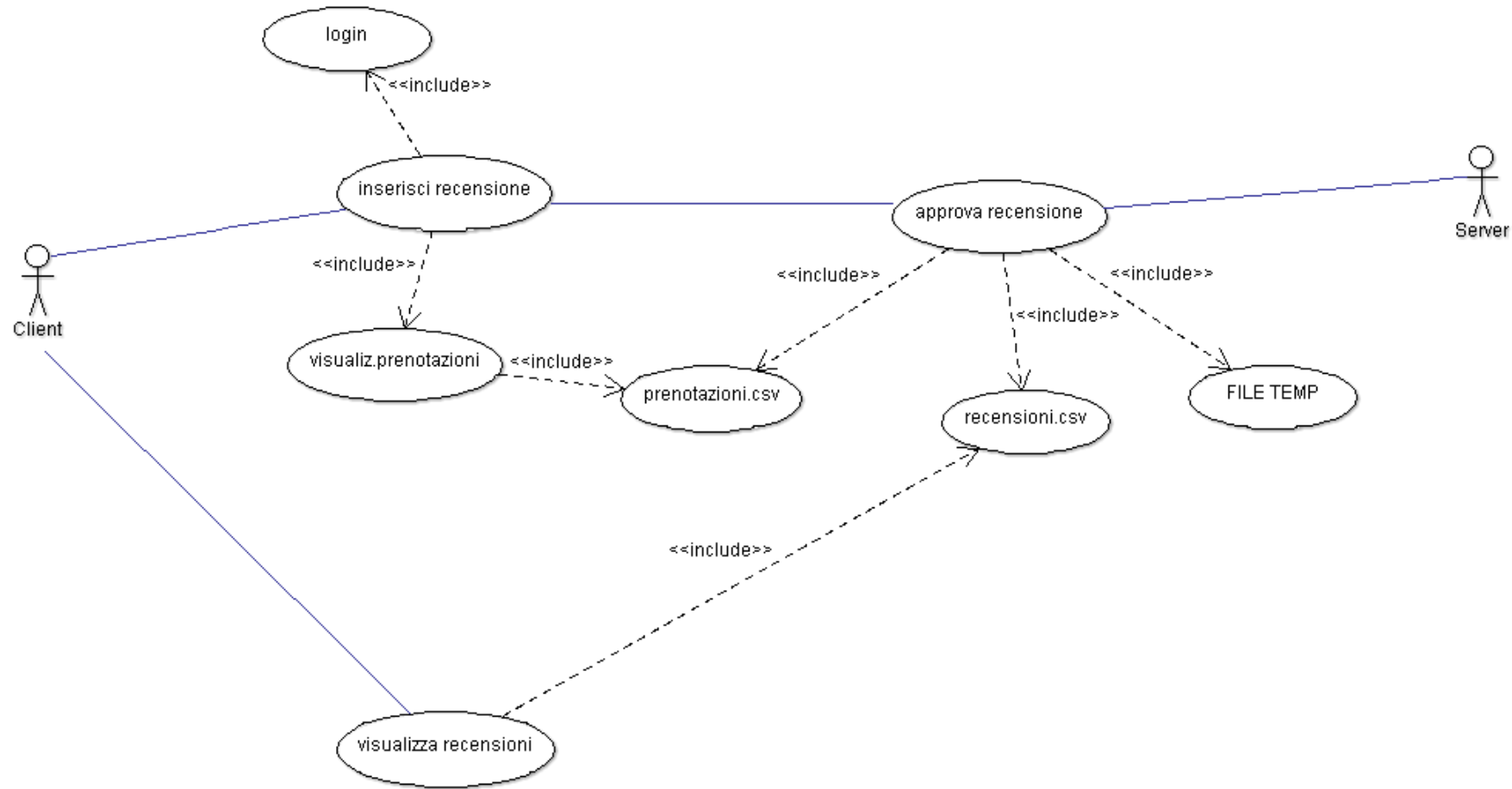
4,3

15,3

7

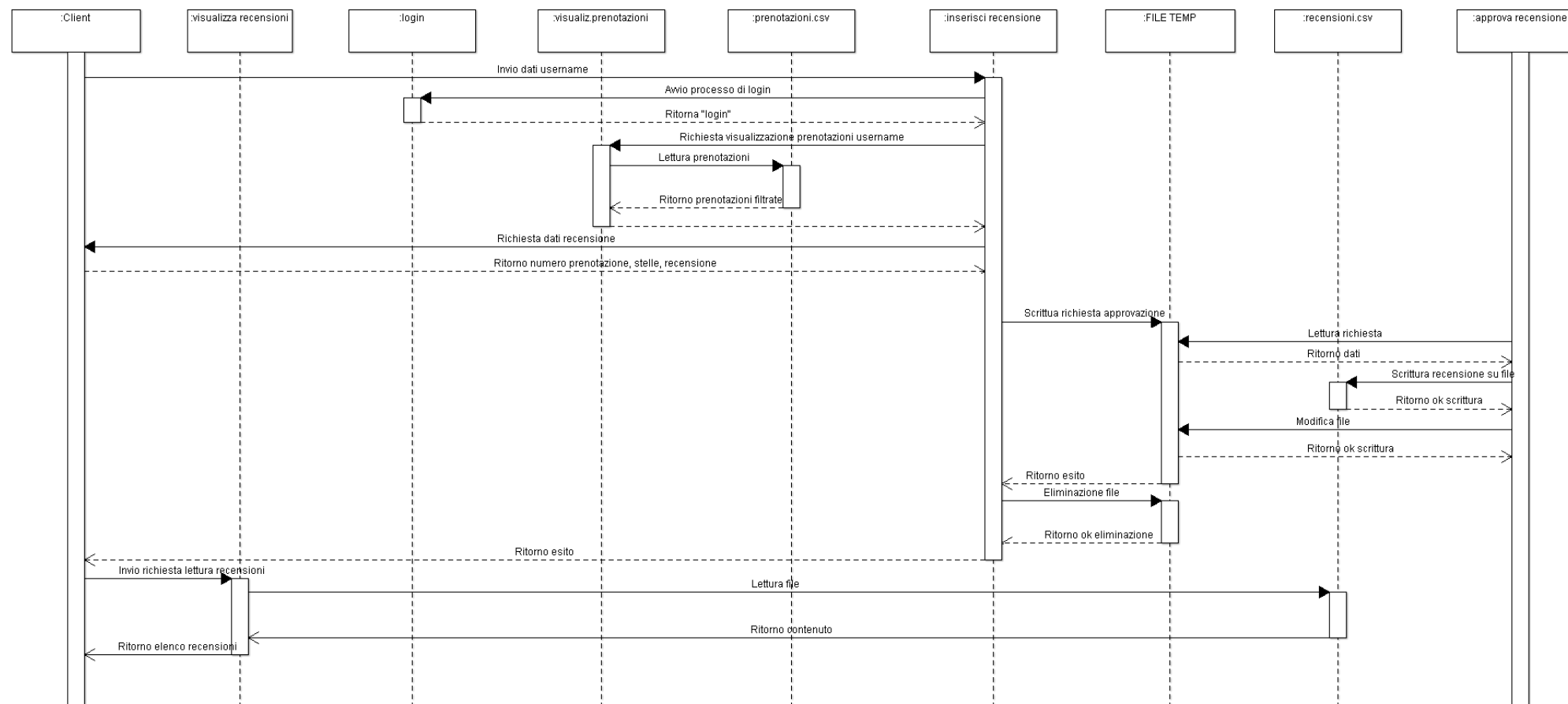
# /\* Analisi dei Requisiti - Esempi di diagrammi UML \*/

// Diagramma di Casi d'uso delle recensioni

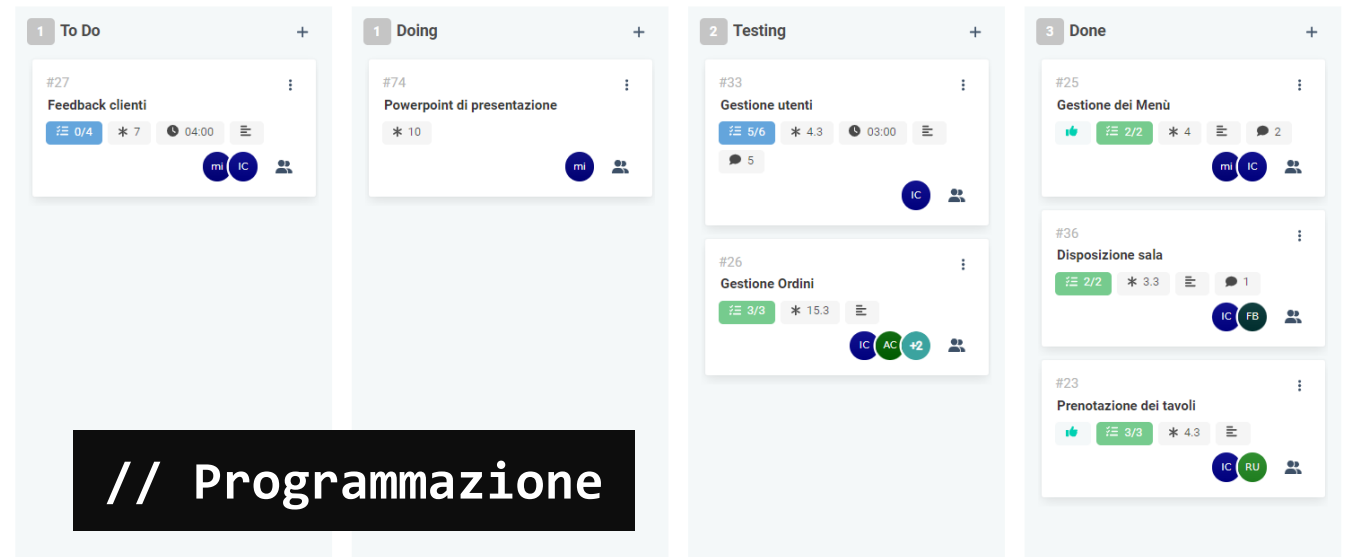
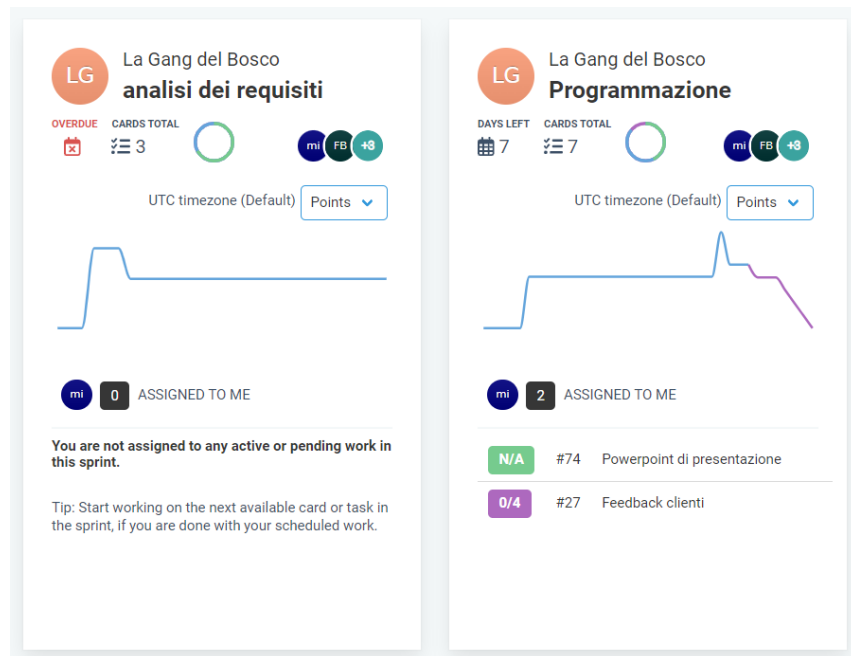


# /\* Analisi dei Requisiti - Esempi di diagrammi UML \*/

// Diagramma di Sequenza delle recensioni



# /\* Sprint Review in FlyingDonut \*/





## /\* Programmazione \*/

// Porzione di codice riguardante la funzione di visualizzazione delle prenotazioni.

```
937 // funzione per visualizzare le prenotazioni di un utente
938 void visualizza_prenotazioni(char* username, char *path){
939
940     // leggiamo il file delle prenotazioni.csv nella cartella path_sala
941     // facciamo un ciclo per leggere le righe del file
942     FILE *file = fopen(path, "r");
943     if(file == NULL){
944         printf("Errore nell'apertura del file\n");
945         return;
946     } else {
947         bool trovato = false;
948         // leggiamo il file e carichiamo ogni riga in una struttura Prenotazione.
949         char riga[MAX_LUNG_PORTATA];
950         // stampiamo le colonne della prenotazione
951         printf("          \tNum\tUser\tData\t\tTavolo\tPosti\n");
952         while(fgets(riga, MAX_PORTATE, file) != NULL){
953             // carichiamo la riga come tipo Prenotazione
954             Prenotazione pren = carica_prenotazioni(riga);
955             // se l'username è uguale a quello passato eliminiamo la prenotazione
956             if (strcmp(pren.username, username) == 0){
957                 printf("Prenotazione \t%d\t%s\t%s\t%d\t%d\n", pren.codice, pren.username, pren.data, pren.tavolo, pren.posti);
958                 trovato = true;
959             }
960         }
961         if (!trovato){
962             printf("Nessuna prenotazione trovata\n");
963         }
964         fclose(file);
965     }
966 }
```

## /\* Programmazione \*/

// Porzione di terminale riguardante il login e la visualizzazione della sala.  
// Sono da fare ulteriori migliorie ai messaggi di errore.

```
[ Client@LaGangDelBosco.it ]: > ./client.exe -v icatalano
Verifica se l'account e' loggato
Inserisci la password: ****
In attesa di risposta
In attesa di risposta
2024-05-23 12:47:25: Account connesso
=====
Account loggato
-----
Dati dell'account:
Nome: Ivan
Cognome: Catalano
Username: icatalano
Email: icatalano@mail.it
Password: ****
Account approvato
-----
Prenotazioni effettuate:

Errore nell'apertura del file
-----
Ordini effettuati:

Errore nell'apertura del file
-----
=====
```

```
[ Client@LaGangDelBosco.it ]: > ./client.exe -sala
Questo comando non ti e' permesso
[ Client@LaGangDelBosco.it ]: > ./client.exe -sala 01/01/2025
In data 01/01/2025 non ci sono prenotazioni.
[ Client@LaGangDelBosco.it ]: > ./client.exe -sala template
Formato data non valido. Inserire la data nel formato gg/mm/aaaa
[ Client@LaGangDelBosco.it ]: > ./client.exe -p_tavolo icatalano 23/05/2024
Controllo se l'utente e' loggato
L'account esiste. Procediamo con la verifica del Login
Account gia' loggato
Non ci sono prenotazioni in data 23/05/2024, la sala e' libera
La disposizione della sala e' la seguente:
La sala non e' piena
Il numero di tavoli e': 4
Il tavolo 1 e' libero, e i posti disponibili sono: 4
Il tavolo 2 e' libero, e i posti disponibili sono: 4
Il tavolo 3 e' libero, e i posti disponibili sono: 4
Il tavolo 4 e' libero, e i posti disponibili sono: 4
Il numero di posti liberi totali e': 16
Tavolo 1:
+++++
+ @ @ @ @ +
+++++

Tavolo 2:
+++++
+ @ @ @ @ +
+++++
```

## `/* Programmazione */`

`// Esempi di bachi riscontrati durante la fase di programmazione`

Se si era loggati con un utente e si voleva visualizzare il profilo di un altro utente, nell'inserimento della password era solo necessario premere INVIO, senza dover digitare alcun carattere, ed il programma permetteva comunque di vedere il profilo dell'altro utente.

Un altro baco simile: se si era loggati con un determinato profilo e si vuole prenotare il tavolo ad un'altra persona, innanzitutto quella persona deve avere un account, poi se si preme INVIO senza loggare al suo account quando si inseriva la password, era comunque possibile prenotare un tavolo a nome di quell'altra persona.

In sostanza, era possibile eseguire il login semplicemente premendo INVIO, senza dover inserire alcun carattere.

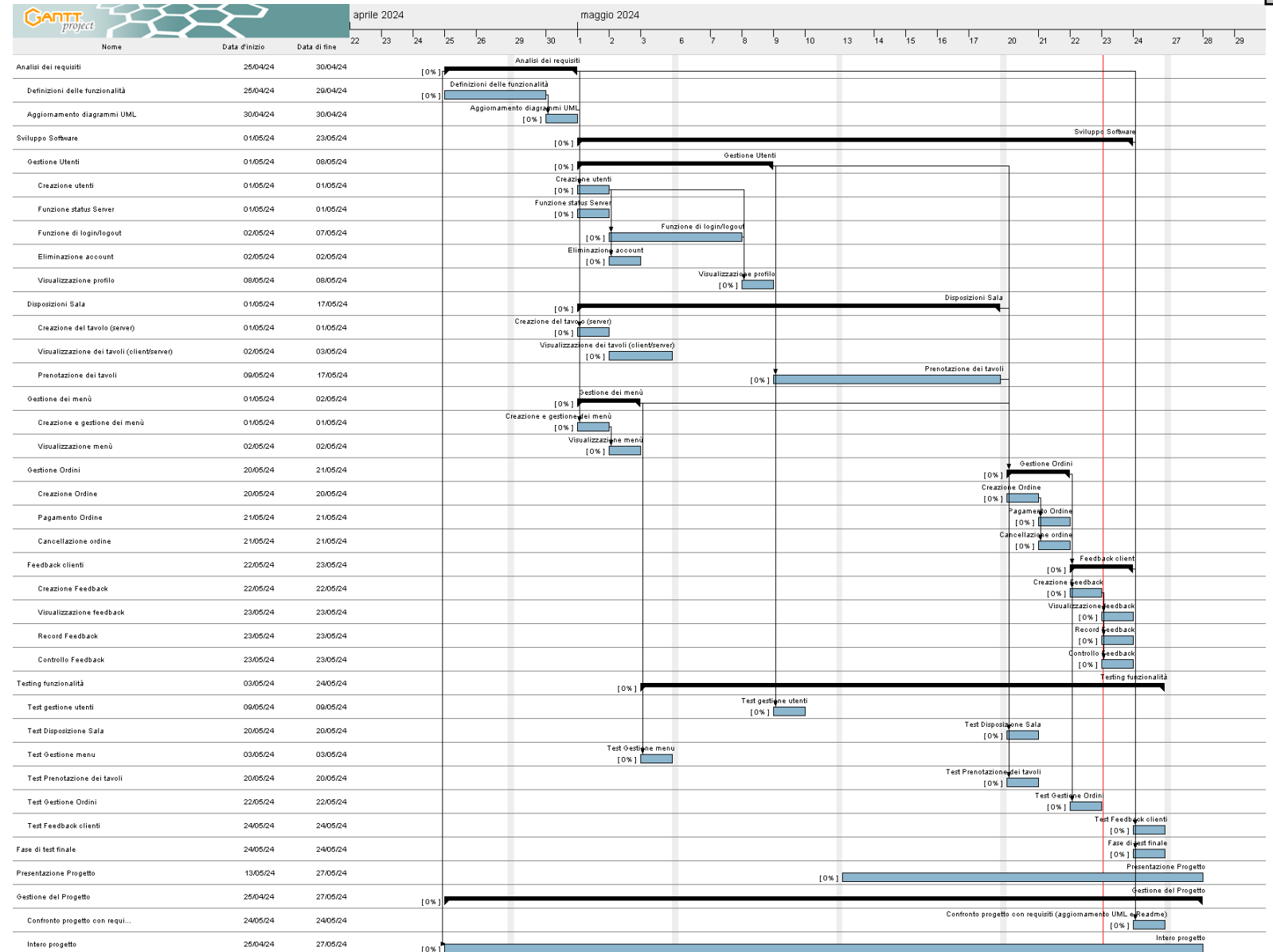
# Diagrammi GANTT

/\*  
\*/

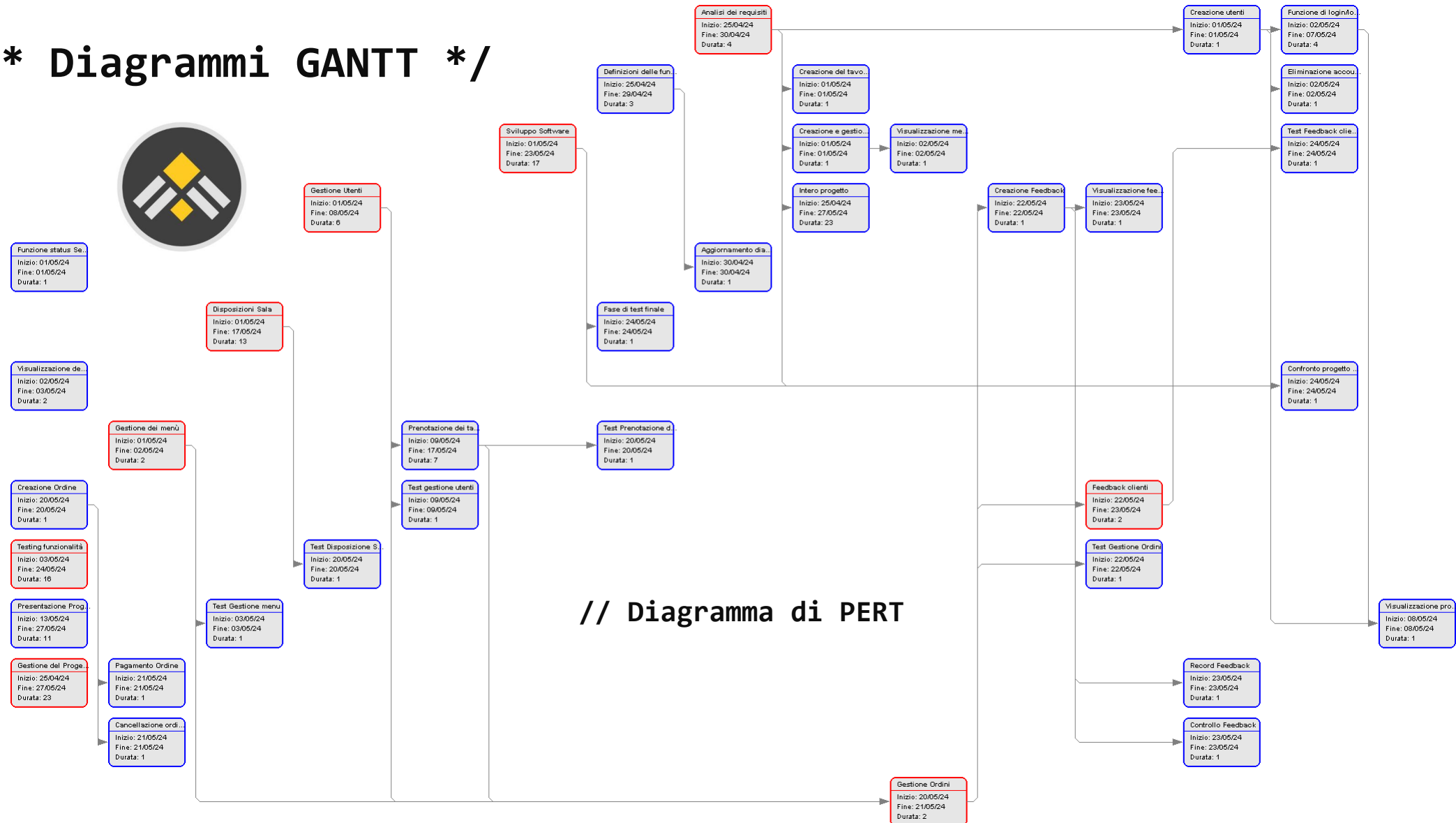


## Organizzazione principale del lavoro

/\*  
\*/



# /\* Diagrammi GANTT \*/

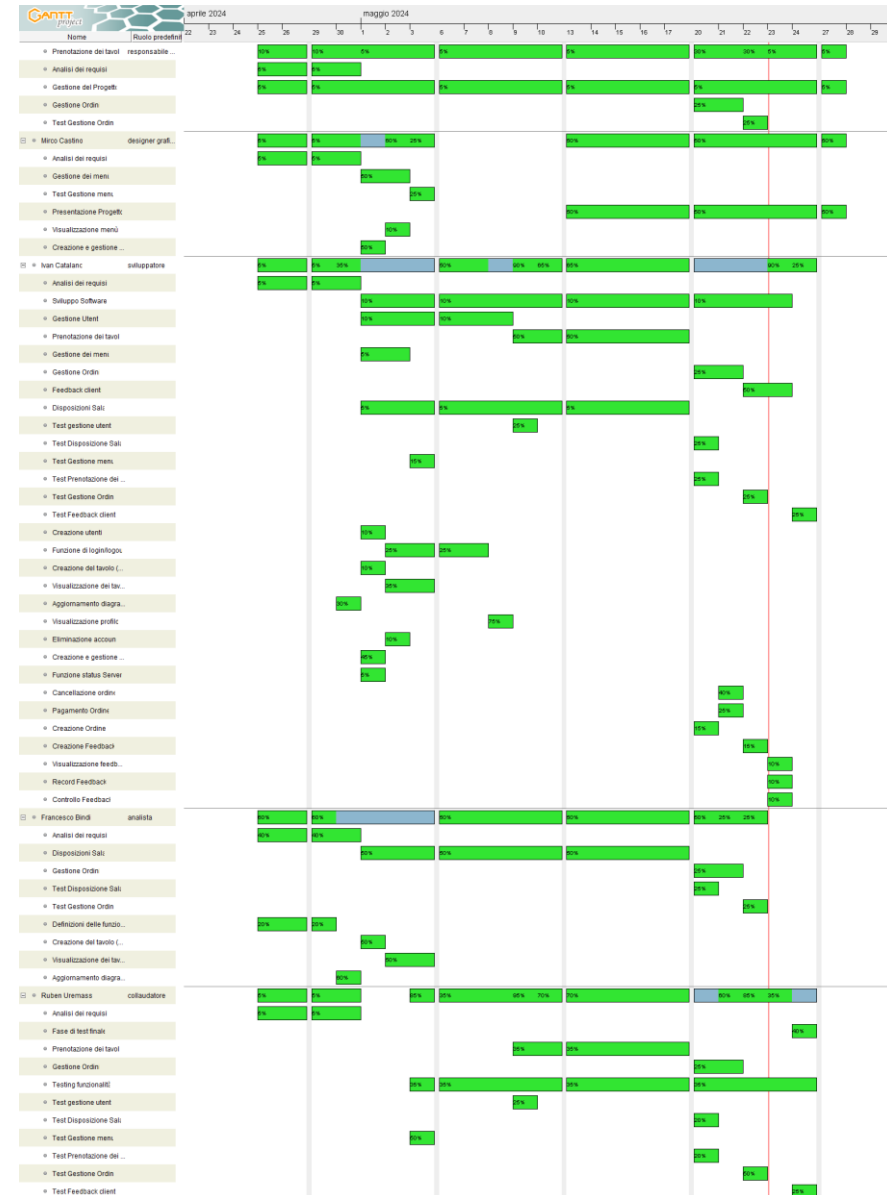


## // Diagramma di PERT

/\* Diagrammi GANTT \*/



// Diagramma delle risorse



/\* README.md \*/

## Guida d'uso

---

### Avvio del Server

La sequenza di avvio dell'app prevede l'avvio del server per il tramite del comando:

```
./server.exe -start
```

Questo comando lancia il server il quale controllerà l'esistenza di tutte le cartelle funzionali al funzionamento del client, creandole in assenza. La seconda operazione che fa è una cancellazione di tutti i file temporanei. Successivamente si mette in ascolto delle richieste che arrivano dal client:

- Richiesta di status (attivo/inattivo) del server
- Approvazione di account appena creati
- Login/Logout utenti
- Richieste di prenotazione di tavoli
- Richieste di ordinazioni delle portate dal menu (con pagamento ed emissione della ricevuta)
- Rilascio di recensioni