# Documentazione

Benedetta Vitale - Emilio Meroni

18 Novembre, 2023

# Contents

# Chapter 1

# Project Plane

## 1.1 Introduzione

Questo progetto verrà svolto da Benedetta Vitale ed Emilio Meroni, entrambi studenti al terzo anno di ingegneria informatica presso l'università di Bergamo.

Il software che andremo a sviluppare è pensato per dare supporto alla attività di ristorazione. In particolare, dovrà assistere alle mansioni dei camerieri, come, prendere le ordinazioni ai tavoli, redigere il conto, trovare tavoli disponibili, ecc.

Abbiamo scelto questa tipologia di sistema dato che Emilio lavora, nei week-end, presso un ristorante, e gli ha incuriosito la gestione interna tramite l'utilizzo dei palmari da parte dei camerieri.

## 1.2 Modello di Processo

Il modello di processo che seguiremo è quello della prototipazione, in particolare la prototipazione evolutiva [Figura: 1.1], molto utile per quanto riguarda la costruzione dell'interfaccia grafica, la parte principale della nostra applicazione.

## 1.3 Organizzazione del Progetto

Utilizzeremo una organizzazione a tre livelli:

1. Livello Data Base
2. Livello Logico
3. Livello Presentazione

Il *livello Data Base*, come da nome, si occuperà della parte del DB, il quale sarà embedded, SQLite. Il *livello Presentazione* sarà quello visto dall'utente che usufruirà dell'applicazione, possiamo definirlo come il livello più esterno dove i dati verranno presentati in modo grafico e la gestione dei tavoli e delle comande

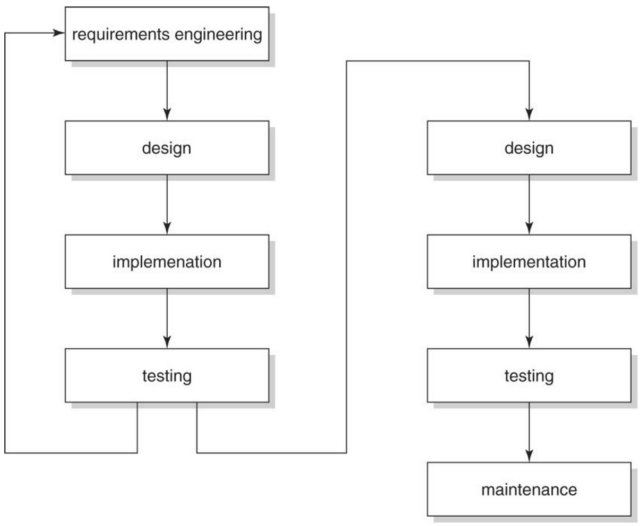


Figure 1.1: Modello Di Processo Evolutivo

è fatto in modo interattivo. L'ultimo livello è quello *Logico*, il quale possiamo posizionarlo graficamente tra il livello Data Base e quello Presentazione, funge da intermediario tra i due livelli e fornisce gli oggetti principali per la gestione dell'applicazione

## 1.4 Standard, Lineeguida, Procedure

Per la parte della stesura della documentazione si è scelto di utilizzare uno tool molto utile nella stesura di documenti professionali, *laTeX*. scelta quasi obbligata derivata dall'utilizzo di *GitHub* insieme al programma di scrittura precendente *Microsoft Word* il quale ha causato difficolta nei "merge" su *GitHub*.

mentre per la parte di *codifica* abbiamo scelto di utilizzare lo standard definito da *Java*[1], in primo luogo perché l'IDE utilizzato per la parte di progammazione è *Eclipse* il quale fornise strumenti per la formatttazione e nominazione di metodi in modo automatico secondo gli standard di *Java*, inoltre, non avendo entrambi molta esperienza ci è venuto comodo utilizzare uno dei pochi standard di codificha che conosciamo.

Lo strumento che utilizzeremo per la condivisione della documentazione e del codice sarà, come già anticipato, la piattaforma di condivisione *GitHub*.

[1]Standard di Java si possono trovare su questo sito

## 1.5 Attività di Gestione

## 1.6 Rischi

In questa sezione si discutono i rischi che potrebbero verificarsi durante lo sviluppo progetto.

Un primo rischio evidenzianto da entrambe le parti sono le difficoltà che potrebbero verificarsi nell'utilizzo di *windows builder*, plugin che consente la progettazione visuale dell'interfaccia utente su *Eclipse*, dovuta da una bassa conoscenza del tool. Un probabile "effetto" sarà quello di allungare i tempi di codifica sul modulo di presentazione.

## 1.7 Personale

Il numero di persone che lavoreranno a questo progetto sarà molto ristretto, nello specifico sono:

- Benedetta Vitale
- Emilio Meroni

I quali participeranno in modo equo a tutte le attività, sia in scrittura della documentazione che sulla parte di codifica. Si è deciso che si lavorerà spesso in coppia, in particolar modo sulla parte di programmazione dato che la parte grafica su *Java* non è mai stata approfondita da entrambe le parti e questo progetto ne ha una parte molto importante.

## 1.8 Metodi e Tecniche

## 1.9 Garanzia e Qualità

Il programma dovrà essere utilizzato in ristoranti, e in particolare dal personale che prenderà le ordinazioni, quindi dovrà essere di *facile* utilizzo, con un focus maggiore sulla funzionalità che sull'estetica. Un'altra qualità che dovrà garantire è la *prevenzione di errori* da parte degli utenti, come ad esempio il "miss-click" (click errati o accidentali). In particolare abbiamo evidenziato quattro qualità che il sistema dovrà possedere:

1. **Semplice**: Il programma sarà scritto per avere poche sezioni scritte e di facile comprensione, anche per chi si approcia al programma per la prima volta. Si utilizzeranno poche schermate che contengono tutto il necessario per le *macro* operazioni definite nelle specifiche dei requisiti e indicate nel case diagram [Figura: 2.1].
2. **Intuiblile**: L'utilizzo dei colori per indicare gli stati dei *tavoli*, in particolre si è deciso che per i *tavoli liberi* si utilizzara il colore verde, per i

*tavoli occupati* il colore rosso e per i *tavoli da pulire* l'arancione [Figura: 1.2].

3. **Prevenzione Errori**: Per la prevenzione degli errori si utilizzeranno schermate *pop-up*, per la convalida degli input

4. **Veloce**: Con questo termine si indica che il programma, che verrà utilizzato da dispositivi touch, dovrà essere principalmente composto da bottoni che minimizzano i tempi di utilizzo, puntando a un uso minimo della tastiera. Comportando un'esperienza più agevole per l'utente.


Figure 1.2: Esempio di colorazione dei bottoni per i tavoli

## 1.10 Workpackages

## 1.11 Risorse

Gli strumenti che verranno adottati in questo progetto saranno:

- Per la parte di codifica, come già detto in precedenza, l'IDE *Eclipse*, uno strumento specifico per la scrittura di codice e gestione di progetti *Java*.

- In scrittura si utilizzerà l'editor di testo open source *VS Code* con l'estensione *LaTeX Workshop* per la scrittura di documenti in formato *LaTeX*. Contribuendo garantire una presentazione accurata dei documenti.

- Infine, riguardo agli strumenti di *Software Configuration Management*, adotteremo *GitHub*. Questa piattaforma sarà impiegata principalmente per la condivisione, gestione e tracciamento delle modifiche al software, oltre che per la documentazione.

## 1.12 Budget

Per quanto riguarda il budget, si è previsto un tempo totale di 80 ore lavorative. con una forte attenzione sulla parte di documentazione, comprendendo anche la parte di documentazione del codice, rispetto a quella di codifica. In prima approssimazione possiamo definire una divisione del tempo come mostrato in figura [1.3].

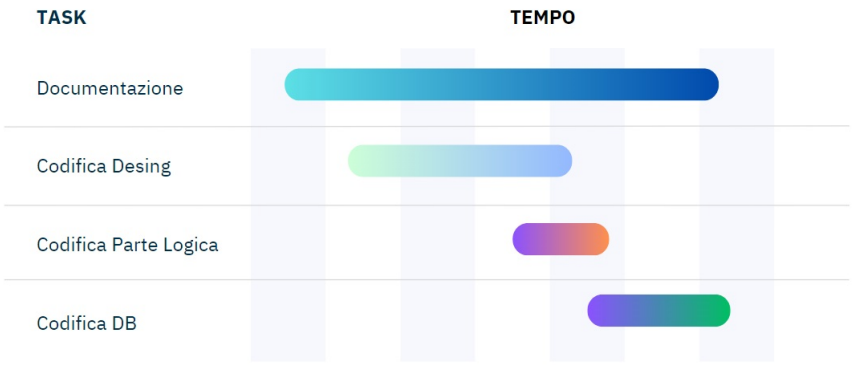


Figure 1.3: Diaramma di Gant

## 1.13 Cambiamenti

## 1.14 Consegna

In questa sezione si discutono i metodi e le scadenze di consegna del progetto, in particolare la consegna si dividerà in due fasi:

- Consegna del *Project Plane*, il quale dovrà essere consegnato circa un mese prima del primo esame scritto, che si svolgerà nel mese di gennaio; quindi, per il mese di dicembre si dovrà effettuare la prima consegna.
- Consegna del *Progetto*, quest'ultimo avrà una scadenza più lunga, infatti, l'ultimo giorno di consegna sarà cinque giorni prima dell'esame orale.

Per quanto riguarda i metodi di consegna si dovrà condividere con il professore Gargantini la repository di github contenente il progetto, per la consegna di questo documento si dovrà indicare nel file reame, della repository, la posizione del project plane. Mentre per la consegna del progetto si dovrà creare un *issue*, intitolata "Approvazione Progetto" ed assegnarla al professore.

# Chapter 2

# Specifiche dei Requisiti

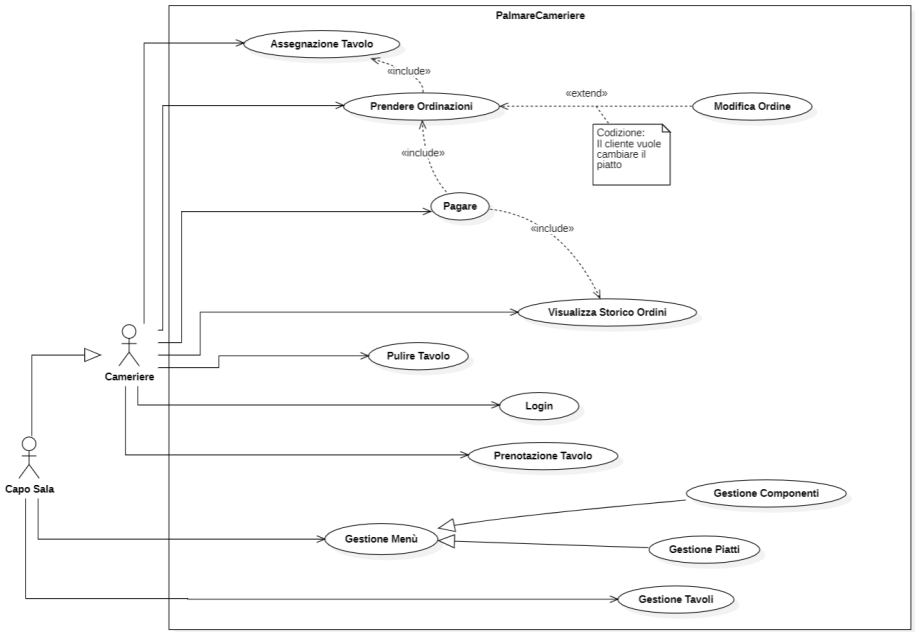


Figure 2.1: Use Case Diagram