Documentazione

Benedetta Vitale - Emilio Meroni

14 dicembre 2023

Indice

Capitolo 1

Project Plan

1.1 Introduzione

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

Il software che andremo a sviluppare è pensato per dare supporto all'attività di ristorazione. In particolare, dovrà assistere alle mansioni dei camerieri, come ad esempio: prendere le ordinazioni, redigere il conto, trovare i 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]. Questo processo è molto utile per quanto riguarda la costruzione dell'interfaccia grafica, dato che ci aiuterà a costruire più velocemente la GUI finale, tramite diversi prototipi d'interfacce utente. La parte principale della nostra applicazione, infatti, sarà la grafica.

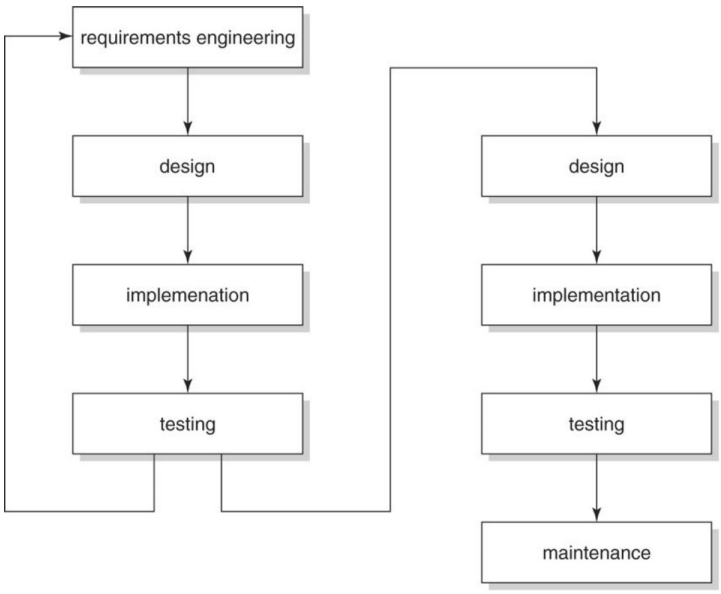

Figura 1.1: Modello Di Processo Evolutivo

1.3 Organizzazione del Progetto

Utilizzeremo una organizzazione a tre livelli:

1. Data Base
2. Logico
3. Presentazione

Il livello *Data Base*, come da nome, si occuperà della parte del DB, il quale sarà embedded.

Il livello *Logico*, 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.

L'ultimo livello, quello di *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 sarà fatta in modo interattivo.

Il progetto verrà suddiviso in:

- **Documentazione**; questa parte verrà svolta da entrambe le figure coinvolte nel progetto.
- **Progettazione desing**; parte eseguita da Benedetta, che apprezza particolarmente questo ambito.
- **Codifica desing**; questa porzione verrà eseguita principalmente da Emilio, dato che ha una maggiore esperienza sulla programmazione.
- **Codifica parte logica** e **data base**; le quali verranno scritte sia da Benedetta e sia da Emilio.

1.4 Standard, Linee guida, Procedure

Per la parte della stesura della documentazione si è scelto di utilizzare un tool molto utile nella scrittura di documenti professionali, *LaTeX*. Scelta quasi obbligata, derivata dall'utilizzo di *GitHub* insieme al programma di scrittura precedente *Microsoft Word*; il quale ha causato difficoltà 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 programmazione è *Eclipse*, il quale fornisce strumenti per la formattazione 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*.

1.5 Attività di Gestione

La priorità fissata per questo progetto sarà quella di avere una documentazione dettagliata, e, specialmente, di avere un project plan completo per l'inizio del mese di dicembre. Parellamente alla stesura di quest'ultimo si è

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

deciso di dedicare, almeno una volta a settimana, del tempo sulla parte di progettazione.

1.6 Rischi

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

Un primo rischio, evidenziato da entrambe le parti, sono le difficoltà che potrebbero verificarsi nell'utilizzo di *windows builder* (plugin che consente la costruzione dell'interfaccia utente su *Eclipse*), dovuta a 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 parteciperanno in modo equo a quasi tutte le attività. Si è deciso che si lavorerà spesso in coppia, in particolar modo sulla parte di programmazione della GUI, dato che la parte grafica su *Java* non è mai stata approfondita da entrambe le parti, e questo progetto ha una parte di grafica molto importante.

1.8 Metodi e Tecniche

Per ogni nuova modifica aggiunta al progetto, prima di eseguire il "merge" su *GitHub*, la verificheremo tramite dei test. Sulla parte grafica il test verrà eseguito in modo visivo, tramite l'esecuzione del programma. Se questo funzionerà ancora si procederà con l'aggiunta di nuove modifiche, così da avere sempre un codice eseguibile.

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 approccia al programma per la prima volta. Si utilizzeranno poche schermate che contengono tutto il necessario per le *macro* operazioni definite.

2. **Intuiblile**: L'utilizzo dei colori per indicare gli stati dei *tavoli*, in particolare si è deciso che: per i *tavoli liberi* si utilizzerà 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.

Figura 1.2: Esempio di colorazione dei bottoni per i tavoli

1.10 Workpackages

I moduli presenti in questo progetto, come già anticipato parzialmente nella sezione 1.3, saranno:

- **Documentazione**
- **Codifica desing**
- **Codifica parte logica**
- **Codifica data base**

L'organizzazione dei pacchetti di lavoro, presenti su *GitHub*, sarà suddivisa in cartelle, per la parte di codifica raggrupperemo i diversi moduli in unico folder; mentre, la documentazione sarà posizionata in una cartella a parte.

Oltre ad avere queste directory, si avrà una parte dedicata ai modelli *ER*(riguardanti il data base) e una per la parte *UML* contenente tutti grafici per la progettazione.

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].

Figura 1.3: Diagramma di Gant

1.13 Cambiamenti

Per quanto riguarda le modifiche che verranno richieste si seguirà la seguente scaletta: inizialmente si verificherà la fattibilità, in caso negativo si cercheranno altre soluzioni che vadano incontro al cliente. Mentre, se si accettano, verrà creato un "branch" sulla piattaforma di *GitHub*. Da qui si aggiornerà la documentazione e i vari diagrammi *UML*, in seguito si procederà alle modifiche del codice. Infine prima di eseguire il "merge" delle modifiche con il *main branch*, si eseguiranno dei test di verifica della funzionalità e di correttezza.

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 Plan**, 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 della documentazione si dovrà indicare nel file *readMe*, della repository, la posizione del project plan. Mentre, per la consegna del progetto si dovrà creare un *issue*, intitolata "Approvazione Progetto" e assegnarla al professore.