

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



Sviluppo software per la gestione di offerte di
fornitura e richieste di acquisto di beni materiali

Tesi di laurea

Relatore

Prof. Tullio Vardanega

Laureando

Riccardo Tassetto

ANNO ACCADEMICO 2019-2020

Sommario

Il presente documento descrive il *stage* da me svolto presso l'azienda San Marco Group S.p.A. nella sede di Marcon (VE), nel periodo che va dal 03-08-2020 al 06-11-2020. L'esperienza di *stage* ha avuto una durata complessiva di 320 ore ed è stata supervisionata e coordinata sia dal mio tutor aziendale, Mauro Vecchiato, che dal mio relatore presso l'ateneo, prof. Tullio Vardanega.

Lo scopo principale di questo progetto di *stage* consisteva nello sviluppo di un sistema di gestione di offerte di fornitura, dedicate a beni materiali e servizi.

In particolare, una volta analizzato il processo aziendale ed essermi interfacciato con gli utenti coinvolti per identificare la soluzione progettuale più adatta, ho progettato un'interfaccia del gestionale utilizzata dagli utenti interni all'azienda e sviluppato un'applicazione *web* utilizzabile dai fornitori per rispondere alle richieste di offerta.

Il presente documento è suddiviso in 4 capitoli:

- **Capitolo 1:** presentazione del contesto aziendale, processi interni e propensione all'innovazione;
- **Capitolo 2:** presentazione dell'offerta di *stage* e motivazioni della scelta effettuata;
- **Capitolo 3:** presentazione dettagliata del progetto, con approfondimento delle sue fasi, delle tecnologie utilizzate e delle soluzioni attuate per realizzarlo;
- **Capitolo 4:** resoconto del lavoro svolto con una valutazione personale sugli obiettivi raggiunti, le difficoltà incontrate e le conoscenze personali e professionali acquisite.

Per la redazione del documento sono state adottate le seguenti norme tipografiche:

- i termini in lingua diversa dall'italiano sono scritti in corsivo;
- i termini del Glossario presenti nel testo sono marcati con una G maiuscola a pedice alla loro prima occorrenza e scritti in corsivo.

Indice

| | | |
|----------|---------------------------------------------|-----------|
| 1 | Analisi del contesto aziendale | 1 |
| 1.1 | L'azienda San Marco Group | 1 |
| 1.2 | Organizzazione del lavoro | 2 |
| 1.2.1 | Assistenza | 3 |
| 1.2.2 | Gestione della configurazione | 3 |
| 1.2.3 | Sviluppo | 4 |
| 1.2.4 | Manutenzione | 5 |
| 1.3 | Propensione all'innovazione | 7 |
| 2 | Progetto di stage | 9 |
| 2.1 | Lo stage per San Marco Group | 9 |
| 2.2 | Contesto attuale | 10 |
| 2.3 | Proposta di stage | 12 |
| 2.3.1 | Prodotti attesi | 12 |
| 2.3.2 | Priorità e obiettivi dello stage | 12 |
| 2.3.3 | Vincoli | 13 |
| 2.3.4 | Pianificazione | 13 |
| 2.4 | Analisi preventiva dei rischi | 14 |
| 2.5 | Motivazione della scelta | 15 |
| 2.5.1 | Motivazioni personali | 15 |
| 2.5.2 | Motivazioni professionali | 15 |
| 3 | Svolgimento dello stage | 17 |
| 3.1 | Analisi dei requisiti | 17 |
| 3.1.1 | Processo aziendale | 17 |
| 3.1.2 | Problematiche attuali | 17 |
| 3.1.3 | Soluzione proposta | 18 |
| 3.1.4 | Requisiti | 19 |
| 3.2 | Progettazione e codifica | 20 |
| 3.2.1 | Tecnologie e strumenti utilizzati | 20 |
| 3.2.2 | Interfaccia del gestionale | 21 |
| 3.2.3 | Web Service | 24 |
| 3.2.4 | Applicazione web | 26 |
| 3.2.5 | Reportistica | 28 |
| 3.3 | Verifica e validazione | 28 |
| 3.3.1 | Verifica | 28 |
| 3.3.2 | Validazione | 30 |
| 3.4 | Consuntivo finale | 30 |

| | | |
|----------|------------------------------------------|-----------|
| 3.4.1 | Copertura di requisiti e test | 30 |
| 3.4.2 | Prodotti realizzati | 31 |
| 4 | Conclusioni | 33 |
| 4.1 | Raggiungimento degli obiettivi | 33 |
| 4.2 | Conoscenze acquisite | 33 |
| 4.2.1 | Conoscenze professionali | 33 |
| 4.2.2 | Conoscenze personali | 33 |
| 4.3 | Valutazione personale | 33 |
| | Glossario | 35 |
| | Acronimi | 37 |

Elenco delle figure

| | | |
|------|------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | Sede principale di Marcon (VE). | |
| | Fonte: https://sanmarcogroup.com/ | 1 |
| 1.2 | Web Help Desk - piattaforma di <i>ticketing</i> utilizzata. | |
| | Fonte: https://www.webhelpdesk.com/ | 4 |
| 1.3 | Esempi di <i>workloads</i> supportati da Visual Studio 2019. | |
| | Fonte: https://visualstudio.microsoft.com | 5 |
| 1.4 | Sintesi del servizio di virtualizzazione offerto da <i>VMware vSphere</i> . | |
| | Fonte: https://docs.vmware.com | 6 |
| 1.5 | Diagramma dell'implementazione di un sistema <i>Fortigate</i> in modalità <i>NAT</i> . | |
| | Fonte: https://docs.fortinet.com | 7 |
| 2.1 | Interfaccia utente <i>Microsoft Dynamics Ax 2009</i> . | |
| | Fonte: https://community.dynamics.com/ | 10 |
| 2.2 | Architettura di sistema di <i>Sage X3</i> . | |
| | Fonte: https://partnerportal.sagex3.com/ | 11 |
| 3.1 | Rappresentazione della pianificazione con diagramma di Gantt. | 18 |
| 3.2 | Immagine di cattura di schermo che mostra il repository del progetto su Gitkraken. | 21 |
| 3.3 | Immagine di cattura di schermo che mostra la rappresentazione di un oggetto <i>X3</i> | 22 |
| 3.4 | Immagine di cattura di schermo che mostra la rappresentazione di una richiesta di offerta. | 22 |
| 3.5 | Immagine di cattura di schermo che mostra la scelta della transazione all'apertura delle richieste di offerta. | 23 |
| 3.6 | Immagine di cattura di schermo che mostra il <i>designer web</i> di <i>X3</i> | 24 |
| 3.7 | Rappresentazione grafica dell'architettura di una installazione di <i>Sage X3</i> . | |
| | Fonte: Documentazione Online Help Center di <i>Sage X3</i> | 25 |
| 3.8 | Immagine di cattura di schermo che mostra i <i>web service</i> creati. | 25 |
| 3.9 | Immagine di cattura di schermo del <i>form</i> di login dell'applicazione <i>web</i> | 27 |
| 3.10 | <i>Form</i> di inserimento risposta nell'applicazione <i>web</i> | 27 |
| 3.11 | Immagine di cattura di schermo che mostra una richiesta dal <i>web service tester</i> di <i>Sage X3</i> | 29 |
| 3.12 | Immagine di cattura di schermo che mostra una risposta dal <i>web service tester</i> di <i>Sage X3</i> | 30 |
| 3.13 | Architettura della soluzione realizzata. | 31 |

Elenco delle tabelle

| | | |
|-----|----------------------------------------------------------------------------|----|
| 2.1 | Tabella riassuntiva degli obiettivi dello <i>stage</i> | 13 |
| 2.2 | Dettaglio pianificazione del progetto di <i>stage</i> | 14 |
| 3.1 | Dettaglio pianificazione aggiornata del progetto di <i>stage</i> | 18 |
| 3.2 | Tabella di riepilogo requisiti. | 19 |
| 3.3 | Tabella di riepilogo requisiti soddisfatti. | 30 |

Capitolo 1

Analisi del contesto aziendale

1.1 L'azienda San Marco Group

San Marco Group è un gruppo aziendale *leader* in Italia nella produzione di pitture e vernici per l'edilizia professionale. Il gruppo con sede principale a Marcon (VE) conta 300 dipendenti, è proprietario di 7 diversi *brand*, ha un fatturato pari a 80 milioni di euro ed una rete distributiva che tocca oltre 100 Paesi. Tutti i prodotti vengono progettati negli stabilimenti italiani, mentre la produzione si divide tra Italia, Bosnia e Russia, con un totale di 8 stabilimenti produttivi che producono per tutti i *brand* del gruppo.

Nella sede di Marcon, che ospita 150 dipendenti, tra impiegati e operai, vi si trovano anche gli uffici delle funzioni centrali del gruppo, le unità produttive e i magazzini. I prodotti San Marco vengono distribuiti attraverso un *network* di rivenditori specializzati in Italia e all'estero, e in esclusiva per il mercato italiano attraverso il primo *marketplace* di proprietà dedicato alle pitture.



Figura 1.1: Sede principale di Marcon (VE).

Fonte: <https://sanmarcogroup.com/>

La gamma di prodotti offerti dall'azienda riesce a soddisfare le richieste di una grande fetta di mercato, dai professionisti nel settore dell'edilizia con prodotti come rivestimenti murali, fissativi e isolanti, idropitture e smalti murali, fino a prodotti per privati come decorativi, prodotti per il legno e per pavimenti. Per realizzare tinte

di colori personalizzate vengono utilizzati strumenti chiamati tintometri elettronici, dispositivi che permettono di miscelare i colori in modo automatizzato, partendo da un colore base e aggiungendone altri seguendo formule precise e producendo un risultato certo, diversamente dalla miscelazione manuale dove piccolissime differenze (nell'ordine dei millilitri) potevano portare a risultati diversi da quelli attesi.

San Marco ha progettato un proprio sistema tintometrico chiamato Marcromie, costituito da un *software*, denominato Leonardo, che si interfaccia con i tintometri e dà accesso ad un archivio di formule sviluppate nel laboratorio di Colorimetria, che permettono di realizzare un'ampia gamma di colori.

Il rivenditore che decide di adottare questo sistema, acquisisce una completa autonomia nella creazione di un nuovo colore, ed è in grado di fornire in tempi rapidi il materiale in tinta.

Oltre alla fornitura del sistema e degli strumenti per utilizzarlo, il reparto di Assistenza Tecnica dell'azienda si occupa dell'installazione, della formazione degli operatori, dell'assistenza colorimetrica e di offrire sempre nuove soluzioni grazie alla continua innovazione di paste coloranti.

1.2 Organizzazione del lavoro

L'ufficio *IT*, che si occupa di gestire l'infrastruttura informatica aziendale, è così strutturato:

- 4 sviluppatori, figure tecniche che si occupano delle attività di sviluppo, assistenza e manutenzione;
- 1 *IT Manager*, responsabile della gestione, manutenzione ed esercizio dei sistemi informativi all'interno dell'azienda;
- 1 *CIO (Chief Information Officer)*, responsabile aziendale delle tecnologie dell'informazione e della comunicazione.

Attualmente le figure tecniche presenti non sono sufficienti per poter adottare un'organizzazione in cui si specializza ogni persona in una delle diverse attività da svolgere, che sono:

- sviluppo *software*;
- analisi e programmazione *Business Intelligence (BI)*_[G];
- sicurezza informatica;
- gestione e manutenzione dell'apparato sistemistico;
- assistenza sul gestionale;
- interrogazioni della base di dati.

Pertanto, l'attuale gestione prevede che tutte le figure tecniche siano formate e si occupino delle attività sopra descritte.

Per sopperire alla mancanza di risorse interne, le attività di gestione e manutenzione dell'apparato sistemistico e di sicurezza informatica sono supportate da aziende esterne. Inoltre, la figura dell'*IT Manager* è fortemente operativa, perché oltre a supervisionare i progetti in corso sia sviluppati internamente che esternamente interfacciandosi con i

fornitori, prende parte attivamente alla gestione dell'apparato sistemistico, dell'assistenza sul gestionale e della programmazione *BI*.

Di seguito sono descritti i principali processi interni e gli strumenti utilizzati a loro supporto.

1.2.1 Assistenza

Il servizio di assistenza è essenziale per supportare gli utenti, tanto gli operai quanto gli impiegati negli uffici, nello svolgimento delle loro mansioni. Visto il bacino di utenza, circa 150 utenti solo nella sede principale di Marcon (VE), l'assistenza di primo livello relativa alle problematiche sia *software* che *hardware* viene fornita da un'azienda esterna, specializzata nel servizio di *help desk*.

La gestione delle problematiche relative al *software* gestionale aziendale rimane compito dei membri dell'ufficio *IT*, che ne conoscono la struttura e sono formati per riuscire ad intervenire.

Strumenti di supporto

Per gestire al meglio le richieste di assistenza interna, viene utilizzato un servizio di *ticketing* chiamato *Web Help Desk* (figura 1.2), che attraverso un portale *web* permette la creazione di richieste da parte degli utenti e una facile gestione delle stesse dai tecnici.

Lo strumento è strutturato in modo da categorizzare le richieste, dare loro un livello di priorità e sulla base di quest'ultimo, assegnarle ad un tecnico che le prenderà a suo carico.

Oltre a fornire supporto nel controllo delle richieste, questo strumento offre funzionalità avanzate come:

- la gestione degli *assets*;
- la creazione di *FAQ* consultabili che raccolgono le soluzioni dei problemi già risolti;
- la gestione del processo di *Change Management* attraverso la creazione di flussi approvativi.

1.2.2 Gestione della configurazione

Per quanto riguarda la gestione della configurazione, tutti i progetti aziendali sono conservati in un *repository* interno, completi di documentazione.

Anche il materiale prodotto durante il tirocinio è stato versionato, in modo da prevenire eventuali perdite di dati e permettere il ripristino in qualsiasi momento di una versione precedente.

Strumenti di supporto

Per la gestione della configurazione si fa affidamento a *Git*, uno strumento di controllo versione distribuito. In particolare viene utilizzato *Github*, servizio di *hosting* che implementa *Git* e facilita lo sviluppo di *software* collaborativo.

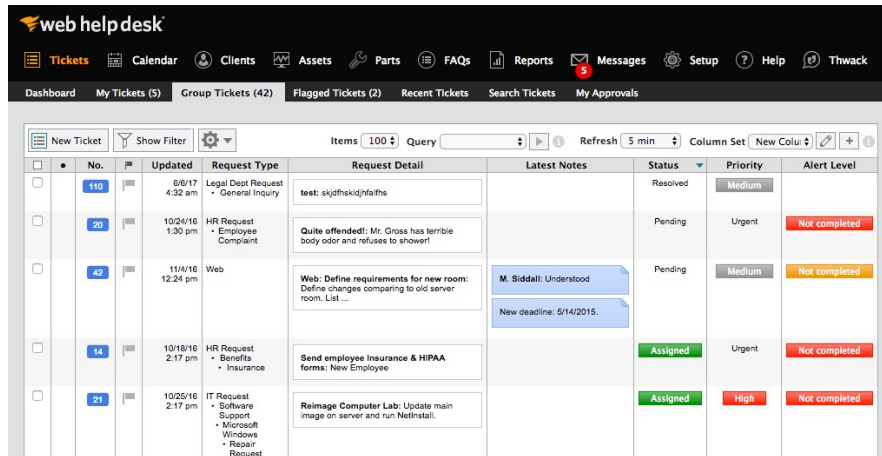


Figura 1.2: Web Help Desk - piattaforma di *ticketing* utilizzata.

Fonte: <https://www.webhelpdesk.com/>

1.2.3 Sviluppo

Per quanto riguarda l'attività di sviluppo, questa comprende:

- sviluppo di nuovi *software*;
- programmazione generica e modifiche su applicativi esistenti.

Gran parte degli applicativi utilizzati in azienda sono stati sviluppati internamente. Prima di iniziare lo sviluppo di un nuovo software, si opera una raccolta dei requisiti attraverso un'analisi preliminare effettuata con i *process owner* delle attività aziendali coinvolte e con gli utenti interni all'azienda che utilizzeranno il *software*, per comprendere le necessità da soddisfare e le problematiche da risolvere.

In seguito, si esegue un breve studio di fattibilità sulla base delle tecnologie da utilizzare e sull'impegno richiesto in termini di tempo.

Successivamente, si sottopone il risultato dell'analisi e dello studio all'*IT Manager* e al *CIO*. Una volta ricevuta l'approvazione, si procede con lo sviluppo.

Per quanto riguarda le richieste di programmazione generica come lo sviluppo di *query* per estrazioni dal database, la progettazione di cruscotti per la *BI*, le modifiche ad alcune funzionalità nel gestionale, lo sviluppo di *web service* per applicativi terzi o le modifiche di piccola entità ad applicativi esistenti, gli sviluppatori riescono a gestire le richieste in autonomia con la sola supervisione dell'*IT Manager*.

Strumenti di supporto

- **Ambiente di sviluppo:** l'ambiente utilizzato per lo sviluppo degli applicativi è *Visual Studio*, il quale dispone di diversi *template* adatti a ciascun linguaggio di programmazione supportato, ad esempio applicazioni *desktop*, libreria di classi, servizi di *Windows* e altri di cui vediamo un esempio alla figura 1.3. Inoltre permette di sviluppare per piattaforme come *Microsoft Azure*, sulla quale si basa il servizio di *Active Directory Azure AD*, il quale viene utilizzato in azienda per la gestione delle identità e l'autenticazione degli utenti;

- **Back-end:** i principali linguaggi utilizzati per lo sviluppo di nuovi *software* sono *C#* e *Java*. Per quanto riguarda la persistenza dei dati è utilizzato principalmente *SQL Server*.
- **Front-end:** viene adottato principalmente il *framework Bootstrap*, perché semplifica la creazione di siti e di applicazioni *web*. Inoltre supporta il *responsive web design*, permettendo che il *layout* delle pagine *web* si regoli dinamicamente, essendoci la necessità di rendere gli applicativi fruibili sia da *desktop* che da dispositivi *tablet* presenti nei reparti di produzione.
- **Programmazione BI:** viene utilizzato *Qlik Sense*, una piattaforma di *data analytics*, che permette di creare soluzioni personalizzate per la *BI* attraverso le *API* messe a disposizione.

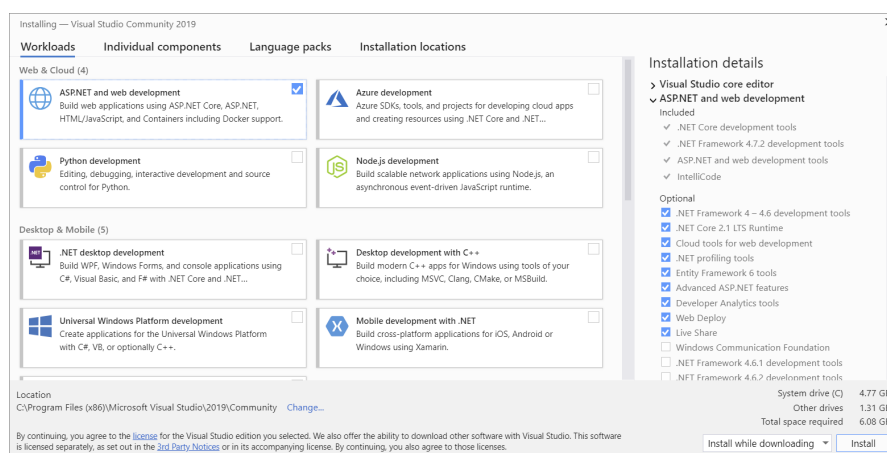


Figura 1.3: Esempi di *workloads* supportati da Visual Studio 2019.

Fonte: <https://visualstudio.microsoft.com>

1.2.4 Manutenzione

Manutenzione del software

Una volta terminato lo sviluppo di un prodotto *software*, viene effettuato un rilascio in un ambiente di *test* per un periodo di due settimane. Vengono effettuati *test* che mettono in luce eventuali difetti del prodotto, in termini di usabilità e prestazioni, in modo da poterli correggere. Successivamente, avviene il rilascio ufficiale in produzione. L'attività di manutenzione viene svolta per tutta la vita del *software* applicando correzioni dove necessario, riadattandolo sulla base dei cambiamenti dell'ambiente di produzione e aggiungendo nuove funzionalità in base alle esigenze degli utenti che lo utilizzano e dei processi che supportano.

Manutenzione sistemistica

Tutti gli aspetti di natura sistemistica sono gestiti internamente, con il supporto di aziende esterne soprattutto nell'ambito della sicurezza informatica e nella gestione

dell'infrastruttura.

Tutti i membri dell'*IT* sono infatti formati in modo tale da supportare i tecnici esterni negli interventi che svolgono nell'infrastruttura aziendale. Le attività principali sono:

- gestione dei *server* virtuali;
- gestione degli apparati di rete (*firewall*, *switch*, *router*, *access point*);
- gestione dei *NAS*, *Network Attached Storage*.

Strumenti di supporto

Per quanto riguarda la gestione dei *server* virtuali, si fa affidamento a *VMware vSphere*, una piattaforma di *cloud computing* per la virtualizzazione.

Questa piattaforma permette di creare più sistemi di computer virtuali, chiamati macchine virtuali (*VM*), le quali vengono eseguite su uno stesso *server* fisico, chiamato *host*.

Attraverso una pagina *web* è possibile collegarsi al *client web vSphere Web Client* e gestire le *VM* tramite le funzionalità messe a disposizione dall'interfaccia.

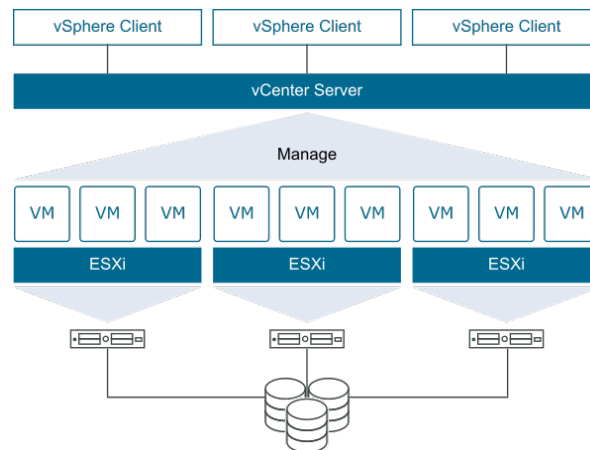


Figura 1.4: Sintesi del servizio di virtualizzazione offerto da *VMware vSphere*.

Fonte: <https://docs.vmware.com>

Fortinet è il *firewall* adottato dall'azienda. La sua gestione avviene attraverso l'interfaccia *web* di amministrazione, che permette di gestire gli utenti abilitati all'accesso alla rete aziendale tramite *Virtual Private Network*_[G], le *policy* che regolano il protocollo *IPv4*, il monitoraggio dell'attività di rete e gli apparati collegati (*switch* e *access point*).

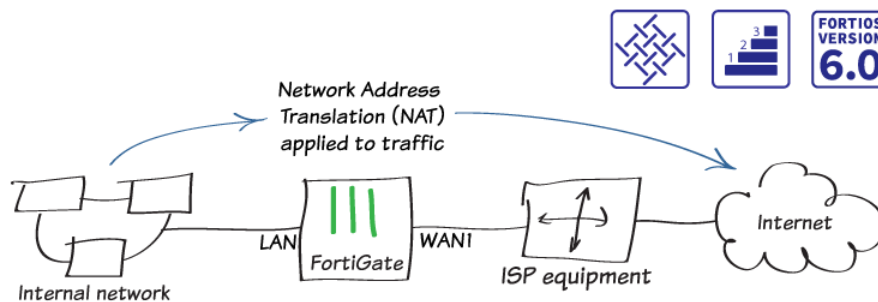


Figura 1.5: Diagramma dell'implementazione di un sistema *Fortigate* in modalità NAT.

Fonte: <https://docs.fortinet.com>

1.3 Propensione all'innovazione

L'azienda si impegna nella ricerca di nuovi strumenti e processi che permettano di migliorare il modo di lavorare portando beneficio ai dipendenti e all'azienda stessa. Questo si manifesta attraverso l'investimento in corsi di formazione per migliorare lo *smart working*, l'introduzione di applicativi come *Microsoft Teams* per incrementare l'efficienza del lavoro di gruppo e l'implementazione di un sistema di comunicazione *VOIP* che permette ai dipendenti l'utilizzo del telefono dell'ufficio attraverso un dispositivo personale mentre sono in *smart working*.

Non sono state prese iniziative solamente per reagire alle necessità della situazione che stiamo vivendo, infatti l'azienda si sta dirigendo sempre di più verso una gestione *cloud* dei servizi e all'utilizzo esteso di *Microsoft Office 365* e di *Sharepoint Online*, una soluzione per la collaborazione e la condivisione di documenti e informazioni, che è andata a sostituire quasi completamente l'archiviazione sui *NAS* utilizzata finora.

Capitolo 2

Progetto di stage

2.1 Lo stage per San Marco Group

Il concetto di *stage* formativo è molto importante per l'azienda San Marco Group. In primo luogo, viene considerata la sperimentazione di nuove tecnologie ed il loro confronto con quelle utilizzate quotidianamente. Per poterlo fare attualmente il personale presente dovrebbe smettere di svolgere le proprie attività, per questo motivo uno stagista universitario diventa una risorsa indispensabile, permettendo quindi all'azienda di sperimentare senza che vengano bloccate le normali mansioni. Ci sono diverse realtà all'interno dell'azienda che possono considerarsi formative per uno studente e che attuano costantemente questa collaborazione:

- **l'ufficio Risorse Umane**, offre a studenti del settore giuridico ed umanistico la possibilità di formarsi dal punto di vista professionale e personale grazie al vasto gruppo eterogeneo di personale presente in azienda;
- **l'ufficio Marketing**, molto sviluppato nell'azienda che dà spazio a figure che hanno intrapreso studi in ambito grafico o orientati alla comunicazione e al *marketing web*;
- **il laboratorio di Ricerca e Sviluppo**, la sezione, presente nella sede di Marcon, affianca studenti del settore chimico ad un gruppo di chimici *senior* che svolge attività di studio e di ricerca su pitture decorative e decorativi per pavimenti, prodotti per le facciate, rivestimenti e sistemi termoisolanti;
- **i Back office**, che comprendono l'ufficio Acquisti e l'ufficio Commerciale, dove studenti del settore economico e commerciale hanno modo di confrontarsi con una realtà molto sviluppata e affermata nel territorio.

In secondo luogo, l'instaurazione del rapporto con l'Università di Padova è dato dalla possibilità di inserimento nell'azienda di risorse derivanti dal mondo accademico che, con una nuova prospettiva data dalle conoscenze apprese e messe in pratica nel corso di studio, possono portare idee creative e innovative, dalle quali può trarre beneficio anche il personale aziendale.

2.2 Contesto attuale

Attualmente l'azienda è nel vivo di un progetto di cambio gestionale, motivato principalmente da:

- la fine del supporto tecnico da parte di *Microsoft* sul gestionale attualmente adottato (*Microsoft Dynamics Ax 2009*, figura 2.1) fissato per il 12 ottobre 2021;
- insoddisfazione da parte della direzione sul prodotto.

Esisteva la possibilità di continuare ad utilizzare la versione 2009 anche oltre alla fine del supporto, affidandosi ad un partner terzo per la manutenzione e le implementazioni dei vari adeguamenti necessari a soddisfare periodicamente le regole stabilite nel contesto fiscale. Questa scelta però è stata ritenuta troppo rischiosa perché, in caso di inadempienze del partner scelto, ci sarebbero potute essere delle conseguenze importanti per l'azienda, come ad esempio l'impossibilità di adempiere agli obblighi fiscali. Si è deciso quindi di cambiare il *software*.

E' stata effettuata l'analisi da parte di un'azienda di consulenza di tutti i processi aziendali, così da individuarne le criticità: in questo modo il cambio di *software* gestionale non avrebbe solamente soddisfatto le necessità ma anche apportato un miglioramento. Successivamente è iniziata la *software selection*.

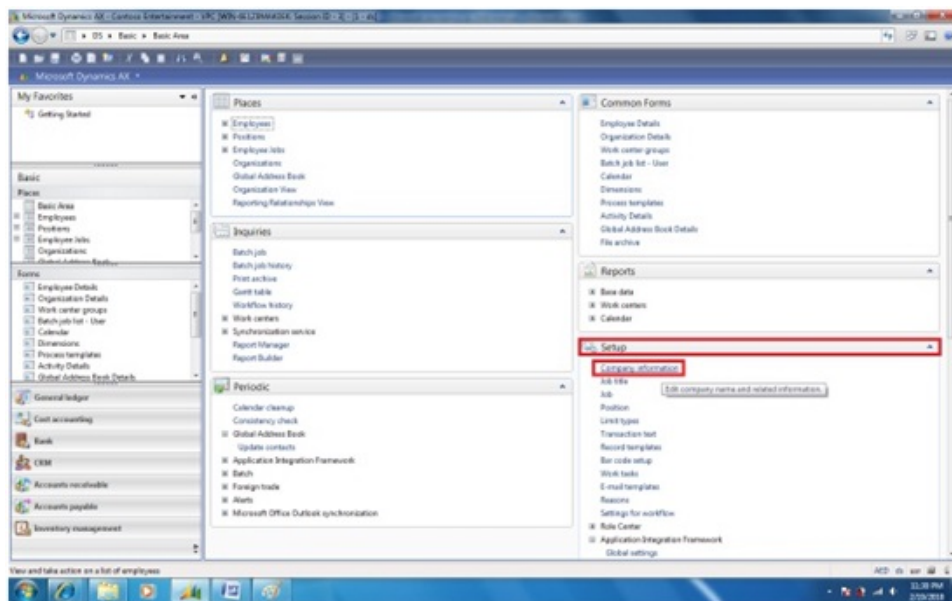


Figura 2.1: Interfaccia utente *Microsoft Dynamics Ax 2009*.

Fonte: <https://community.dynamics.com/>

Tra i vari *software* gestionali presenti sul mercato, è stato preso in considerazione *Microsoft Dynamics 365*. La nuova versione dell'ERP di *Microsoft* differisce in maniera sostanziale da quella attualmente utilizzata, principalmente per quanto riguarda:

- la re-ingegnerizzazione della base di dati, che richiederebbe una reimpostazione radicale della base di dati attuale;

- la dismissione del modello di sviluppo a *layer*, che consiste nell'utilizzo di una gerarchia di livelli nei quali vengono implementati i metodi degli elementi applicativi presenti nel gestionale. Quando viene eseguito un metodo di un qualsiasi elemento applicativo, il sistema parte dal *layer* più esterno per verificare ed eseguire, se disponibile, un'implementazione del metodo per quell'elemento, altrimenti si sposta sul *layer* più interno per ripetere l'operazione fino ad arrivare al *layer* di sistema. In questo modo è possibile personalizzare facilmente le varie funzionalità presenti, attività che con la nuova versione richiederebbero un processo più impegnativo.

L'adozione di *Microsoft Dynamics 365* sarebbe stata troppo impegnativa dal punto di vista tecnico sia nel breve che nel lungo termine, per questo motivo è stato scartato. Alla fine della valutazione la scelta è ricaduta su *Sage X3*, per la sua adattabilità e perché rispecchia maggiormente le necessità aziendali; è perciò iniziata la fase di implementazione del nuovo *software* attraverso:

- sviluppi per adeguamento delle funzionalità del nuovo gestionale agli *standard* aziendali;
- migrazione dei dati;
- formazione degli utenti finali.

Di seguito vediamo un'immagine riassuntiva sulle tecnologie che compongono il software gestionale e un'accenno alla sua architettura di sistema, descritta più in dettaglio in 3.2.2.

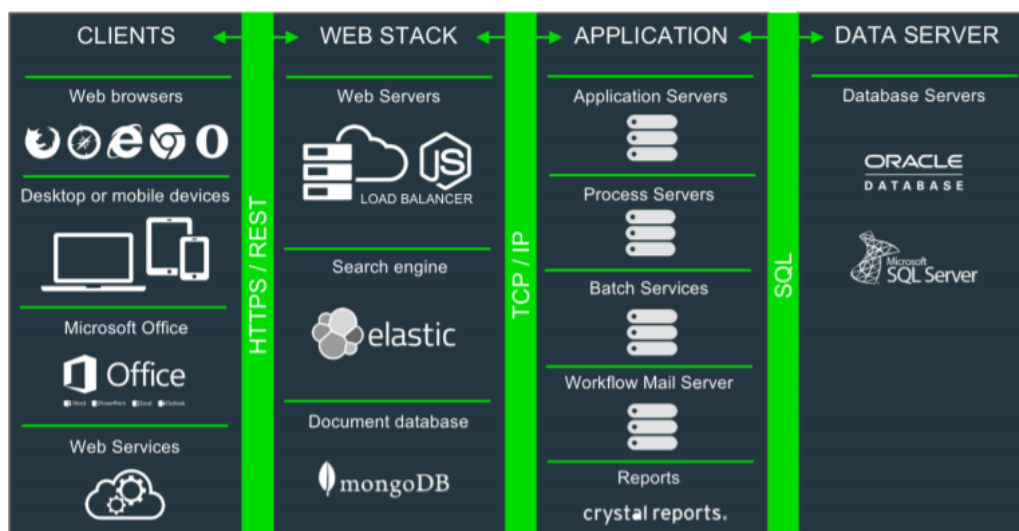


Figura 2.2: Architettura di sistema di *Sage X3*.

Fonte: <https://partnerportal.sagex3.com/>

2.3 Proposta di stage

Lo scopo dello *stage* consisteva nello sviluppo *software* di un sistema di gestione di offerte di fornitura, dedicate a beni materiali e servizi. Dopo l'analisi del processo, il mio compito era quello di sviluppare un'applicazione *web* che permettesse agli utenti interni la gestione delle richieste e ai fornitori designati di rispondere con delle proposte di offerta. L'applicazione da sviluppare doveva avere le seguenti caratteristiche:

- capacità di interfaccia con il database del gestionale tramite *web service*;
- la sezione destinata alle richieste di servizio di trasporto doveva rispettare il layout del modulo cartaceo utilizzato;

Inoltre avevo il compito di redigere un *report* con il dettaglio della richiesta e in base al tempo disponibile, fornire delle statistiche sulla base dei dati utilizzati. Quest'ultimi sarebbero stati utili all'analisi di *BI* in un'ottica di miglioramento del processo aziendale.

2.3.1 Prodotti attesi

L'attività di *stage* prevedeva la produzione dei seguenti oggetti, documenti e *software*:

- relazione sul processo aziendale coinvolto;
- relazione sulla progettazione architettuale;
- codice sorgente dell'applicazione *web*;
- manuale e documentazione riguardante la struttura dell'applicazione *web* per manutenzione ed eventuali integrazioni;
- *report* di dettaglio dell'offerta;
- *report* con statistiche utili all'analisi di *BI*.

2.3.2 Priorità e obiettivi dello stage

Per identificare gli obiettivi ed il loro livello di priorità, ho associato a ciascuno di essi un codice nel formato sottostante:

<obiettivo.[sotto-obiettivo]-tipologia>.

Il significato del codice è il seguente:

- **obiettivo**, indica il numero univoco dell'obiettivo;
- **sotto-obiettivo**, è il numero univoco del sotto-obiettivo, ed è opzionale, essendo riportato solo nel caso in cui l'obiettivo sia suddiviso in sotto-obiettivi;
- **tipologia**, indica il livello di priorità assegnato all'obiettivo.

La tipologia determina l'importanza di realizzare tale obiettivo, ed è scelta tra le seguenti:

- **OB**, indica un obiettivo obbligatorio, il cui raggiungimento è necessario;
- **DE**, indica un obiettivo desiderabile, il cui raggiungimento è importante e dal riconoscibile valore aggiunto, ma non necessario.

Tabella 2.1: Tabella riassuntiva degli obiettivi dello *stage*.

| Obiettivo | Descrizione |
|-----------|---------------------------------------------------------------------------|
| 1-OB | Comprensione del problema |
| 2-OB | Comprensione delle tecnologie di <i>backend</i> |
| 2.1-OB | Comprensione di <i>Sage X3</i> |
| 2.2-OB | Comprensione del database <i>SQL Server</i> |
| 2.3-OB | Comprensione della tecnologia dei <i>web service</i> |
| 3-OB | Comprensione delle tecnologie di <i>frontend</i> |
| 3.1-OB | Comprensione di <i>Javascript</i> |
| 3.2-OB | Comprensione di <i>jQuery</i> |
| 4-OB | Sviluppo dell'applicazione <i>web</i> |
| 5-OB | Sviluppo del <i>report</i> di dettaglio della richiesta |
| 6-OB | Validazione del prodotto |
| 7-DE | Sviluppo del <i>report</i> con le statistiche utili all'analisi <i>BI</i> |

2.3.3 Vincoli

Vincoli temporali e organizzativi

Il progetto di *stage* ha avuto una durata di 320 ore, distribuite nell'arco di tre mesi, con settimane lavorative di 25 ore ciascuna. In accordo con il relatore e il *tutor* aziendale, lo *stage* è stato svolto prevalentemente in modalità *smart working*. Con lo scopo di informare il mio relatore dei progressi, ogni 5 giorni lavorativi dovevo provvedere a inviare un resoconto dello stato di avanzamento rispetto alle attese del piano di lavoro e ad eventuali deviazioni da esso.

Invece, con il *tutor* aziendale, avevo previsto almeno un incontro a settimana in remoto o fisicamente in azienda per discutere del progresso e di eventuali cambiamenti da apportare al progetto.

Vincoli tecnologici

Lo sviluppo del progetto prevedeva alcuni vincoli imposti dall'azienda. In primo luogo l'utilizzo della tecnologia *SOAP*, *Simple Object Access Protocol*, per i *web service* che permettevano l'interfacciamento tra l'applicazione *web* e il gestionale. In secondo luogo, l'utilizzo dello strumento per la produzione di stampe di dati *Crystal Reports* per produrre il *report* di dettaglio della richiesta d'offerta e il *report* con le statistiche utili all'analisi della *BI*.

2.3.4 Pianificazione

Il Piano di Lavoro, redatto insieme al *tutor* aziendale, prevedeva inizialmente 4 fasi, ognuna delle quali della durata di 3 settimane lavorative:

- studio del processo aziendale e formazione sulle tecnologie coinvolte nello sviluppo dell'applicazione *web* e dell'interfacciamento con il gestionale;
- progettazione della soluzione e configurazione dell'ambiente di sviluppo;
- realizzazione dell'applicazione *web* e fase di *testing*;

- formazione sul *software* per la produzione di stampe e sviluppo dei *report*;

L'ultima settimana di *stage* prevedeva la verifica e la validazione finale. Di seguito un dettaglio delle fasi appena descritte:

Tabella 2.2: Dettaglio pianificazione del progetto di *stage*.

| Ore | Attività |
|-----|------------------------------------------------------------|
| 40 | Formazione sulle tecnologie |
| 35 | Analisi del problema |
| 25 | Stesura documentazione relativa ad analisi e progettazione |
| 55 | Progettazione della soluzione |
| 15 | Configurazione dell'ambiente di sviluppo |
| 60 | Sviluppo dell'applicazione <i>web</i> |
| 20 | Test sull'applicazione <i>web</i> |
| 25 | Test sui <i>web service</i> |
| 25 | Formazione su strumento e produzione <i>report</i> |
| 5 | Stesura documentazione finale |
| 15 | Verifica e validazione finale |

2.4 Analisi preventiva dei rischi

Nella fase iniziale di analisi ho individuato alcuni possibili rischi in cui poter incorrere.

Successivamente, ho provveduto ad elaborare delle possibili soluzioni per far loro fronte.

1. Tecnologie e *framework* sconosciuti

Descrizione: Durante lo svolgimento dello *stage* era previsto l'utilizzo di tecnologie a me ancora sconosciute.

Soluzione: Nella fase di pianificazione iniziale ho programmato un periodo di auto-formazione sulle tecnologie e sui *framework* da utilizzare.

2. Incomprensioni sul processo aziendale

Descrizione: A causa dei vari enti coinvolti nella fase di analisi e del gran numero di attività svolte attualmente extra sistema (utilizzo di moduli cartacei, contatti telefonici direttamente coi fornitori, ecc.), avevo previsto di non riuscire a delineare e comprendere pienamente il processo aziendale interessato.

Soluzione: In accordo con il *tutor* aziendale, sono stati individuati dei *key users*, uno per ogni ufficio coinvolto, che si sono resi disponibili ad offrire un maggior dettaglio sulle fasi che interessano la loro attività.

3. Scelte non ottimali

Descrizione: A causa dell'inesperienza avevo considerato l'idea di non riuscire a comprendere le attività da svolgere o di fare scelte non adeguate alle aspettative aziendali.

Soluzione: Tutte le scelte progettuali fondamentali sono state discusse e supervisionate dal *tutor* aziendale.

2.5 Motivazione della scelta

2.5.1 Motivazioni personali

Una delle principali motivazioni che mi hanno portato a scegliere questa proposta di *stage* curricolare, era il contesto di inserimento: volevo capire come una grande azienda, il cui settore di impiego non è strettamente legato a quello informatico, si approccia alle nuove tecnologie e si appoggia all'informatica per il suo business.

Il mio intento era di riuscire ad avere una visione d'insieme per capire quali attività vengono svolte per supportare un'azienda, non solo del contesto dello sviluppo *software* ma anche nell'ausilio di altri processi.

Inoltre, l'opportunità offertami dall'azienda non si sarebbe fermata al semplice *stage* curricolare o all'assunzione per la sola occupazione lavorativa, ma quella che mi è stata proposta era allo stesso tempo la possibilità di supporto nel mio percorso di studi fornendomi un impiego retribuito, a riprova di una *vision* contemporanea che vede negli stagisti non uno svantaggio, ma una risorsa che porta nuova linfa all'interno di un mondo lavorativo consolidato.

A conclusione dello *stage* mi attirava la possibilità di continuare a seguire le attività successive allo sviluppo del prodotto, come la formazione degli utenti, e vedere i risultati del mio lavoro nel lungo termine.

2.5.2 Motivazioni professionali

La mia scelta di *stage* è ricaduta su questa azienda perché rispetto alle altre con cui ho avuto contatti grazie all'evento di incontro *StageIt* in cui le principali aziende del territorio propongono i loro *stage* formativi in ambito *Information and Communications Technology (ICT)*_[G], questa mi avrebbe offerto l'opportunità di vedere non solo da vicino il processo di sviluppo di un nuovo gestionale, ma di svilupparlo a diretto contatto con gli utenti finali e non solo come un servizio offerto dall'esterno (ad esempio un'azienda di sola consulenza informatica), esperienza molto più stimolante per chi alle prime armi si avvicina al mondo del lavoro.

Mi interessava dunque il fatto di potermi rapportare direttamente con gli utenti finali del prodotto, in modo da poter comprendere meglio le problematiche che si possono incontrare in fase di analisi e riuscire ad applicare i concetti studiati nel corso di Ingegneria del Software.

Infine, essendo in pieno sviluppo un progetto di cambio gestionale, ho ritenuto fosse un'opportunità ad alto contenuto formativo, sia per le tecnologie utilizzate a me ancora sconosciute, sia per la possibilità di interfacciarmi e collaborare direttamente con figure professionali interne ed esterne all'azienda impegnate nel progetto.

Capitolo 3

Svolgimento dello stage

3.1 Analisi dei requisiti

3.1.1 Processo aziendale

Attualmente vengono gestite due tipologie di richieste di offerta:

- acquisto di beni e servizi;
- richieste di trasporto merci.

Le richieste di acquisto di beni e servizi (come servizi di pulizia, di vigilanza ecc.) vengono ad oggi interamente gestite all'interno del gestionale. Le richieste di trasporto merci, invece, vengono effettuate attraverso la compilazione di un modulo cartaceo (chiamato MPQ), contenente tutte le caratteristiche dell'offerta; il modulo è veicolato tra i diversi uffici che ne compilano le sezioni di competenza:

- **ufficio Commerciale**, che inserisce i dati necessari alla consegna;
- **ufficio Logistica**, che inserisce le informazioni sul carico e il trasporto;
- **ufficio Acquisti**, che perviene il modulo compilato e lo invia ai fornitori designati.

Una volta ricevute le risposte, l'ufficio acquisti designa il vincitore e lo comunica all'ufficio commerciale, che contatta il cliente per l'accettazione o il rifiuto dell'offerta.

3.1.2 Problematiche attuali

Le comunicazioni tra uffici e fornitori sono effettuate telefonicamente o via *e-mail* e non vi è traccia all'interno del gestionale di alcun dato (numero di richieste effettuate, numero di richieste accettate/respinte, ecc.) ad eccezione del prezzo dell'offerta del fornitore vincitore della gara, che viene inserito manualmente dall'ufficio acquisti. La soluzione proposta inizialmente dall'azienda, ovvero la creazione di un'applicazione *web* utilizzabile sia dagli utenti che dai fornitori, non sarebbe andata incontro pienamente alle esigenze degli utenti. Quest'ultimi infatti avrebbero dovuto utilizzare un'applicazione esterna all'ambiente del gestionale. Questo avrebbe comportato:

- ulteriore formazione per gli utenti;
- spostamento tra ambienti diversi (gestionale e applicazione *web*), perdita di tempo nel passaggio dall'uno all'altra rispetto all'utilizzo del solo gestionale.

Oltre a questo, il nuovo gestionale *Sage X3* presenta una funzionalità di gestione delle richieste di offerta di default, fortemente personalizzabile. Ecco quindi, che in accordo con il *tutor* aziendale, ho quindi ridefinito la soluzione rispetto a quella inizialmente proposta dall'azienda.

3.1.3 Soluzione proposta

Vista la funzionalità offerta da *Sage X3* e i limiti della prima soluzione proposta dall'azienda, ho ritenuto opportuno ridefinire la soluzione personalizzando all'interno di *Sage X3* il modulo di gestione delle richieste di offerta, che verrà utilizzato dagli utenti interni. L'applicazione *web*, che sarà pertanto utilizzata solo dai fornitori per rispondere alle richieste di offerta, comunicherà con il gestionale tramite *web service SOAP*. Questa modifica ha richiesto un aggiornamento delle attività e della pianificazione dello *stage*, come segue:

Tabella 3.1: Dettaglio pianificazione aggiornata del progetto di *stage*.

| Ore | Attività |
|-----|------------------------------------------------------------|
| 55 | Formazione sulle tecnologie |
| 35 | Analisi del problema |
| 25 | Stesura documentazione relativa ad analisi e progettazione |
| 55 | Progettazione della soluzione |
| 15 | Configurazione dell'ambiente di sviluppo |
| 20 | Sviluppo dell'interfaccia nel gestionale |
| 5 | Test su interfaccia gestionale |
| 35 | Sviluppo dell'applicazione <i>web</i> |
| 15 | Test sull'applicazione <i>web</i> |
| 15 | Test sui <i>web service</i> |
| 25 | Formazione su strumento e produzione <i>report</i> |
| 5 | Stesura documentazione finale |
| 15 | Verifica e validazione finale |

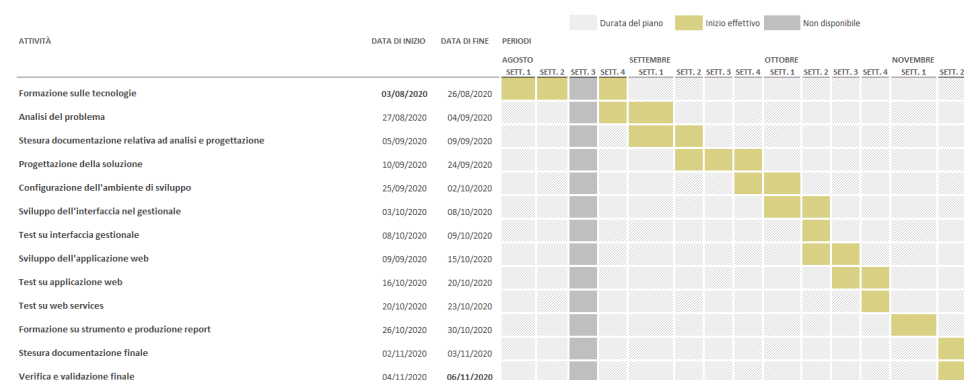


Figura 3.1: Rappresentazione della pianificazione con diagramma di Gantt.

Come si evince dal dettaglio nella tabella 3.1, sono aumentate sensibilmente le ore dedicate alla formazione, necessarie ad apprendere le nozioni fondamentali per sviluppare l'interfaccia nel gestionale. Sono diminuite le ore necessarie allo sviluppo dell'applicazione *web*, in quanto la gestione delle richieste da parte degli utenti aziendali viene fatta all'interno di *Sage X3* e sono state aggiunte le ore necessarie all'implementazione e ai *test* dell'interfaccia del gestionale.

3.1.4 Requisiti

I requisiti che ho individuato sono stati classificati, assieme al *tutor* aziendale, per tipologia:

- **funzionali**, che individuano tutte le funzionalità previste dall'applicazione *web* e dall'interfaccia in *Sage X3*;
- **qualitativi**, in cui ho inserito l'utilizzo del sistema di versionamento con *Github*;
- **di vincolo**, in cui rientrano i vincoli sulle tecnologie da utilizzare nello sviluppo del progetto, ossia *framework* e *web service*;
- **prestazionali**, non è risultato necessario definire alcun vincolo necessario a garantire alcuna prestazione.

Oltre alla tipologia, i requisiti vengono suddivisi anche per importanza:

- **obbligatorî**, necessari alla realizzazione del progetto;
- **desiderabili**, non necessari ma se soddisfatti portano valore aggiunto al prodotto.

Tabella 3.2: Tabella di riepilogo requisiti.

| Tipo | Obbligatorio | Desiderabile | Totale |
|----------------------|--------------|--------------|--------|
| Funzionale | 25 | 1 | 26 |
| Qualitativo | 1 | 0 | 1 |
| Vincolo | 3 | 0 | 3 |
| Prestazionale | 0 | 0 | 0 |
| Totale | 29 | 1 | 30 |

Ritengo che la quantità di requisiti individuati sia stata congrua con il tempo che ho impiegato per soddisfarli, anche se per i requisiti che riguardavano i *web service* ho impiegato più tempo del previsto, sia per la tecnologia su cui mi dovevo formare, sia per problematiche esterne.

Grazie all'esperienza maturata nel progetto svolto durante il corso di Ingegneria del Software, ho appreso che la maggior parte dei requisiti non è individuabile facilmente e non è sufficiente basarsi sul capitolato presentato dall'azienda.

Per questo motivo ho cercato di approfondire il più possibile la fase di analisi, soprattutto attraverso colloqui con tutti gli utenti interessati, in modo da comprendere a fondo le necessità dell'azienda e le caratteristiche che doveva avere la mia soluzione.

3.2 Progettazione e codifica

3.2.1 Tecnologie e strumenti utilizzati

Visual Studio

Visual Studio nella sua versione 2019 è l'*IDE* che ho utilizzato come ambiente di sviluppo per la creazione dell'applicazione *web*. L'*IDE* incorpora la versione 4.7.2 di *.NET Framework*, ambiente di esecuzione *runtime* della piattaforma tecnologica *.NET*. Ho scelto questo strumento in primo luogo perché è l'ambiente utilizzato maggiormente all'interno dell'azienda, per la sua compatibilità ed estendibilità con i vari applicativi e servizi *Microsoft* presenti nell'infrastruttura interna. Inoltre dispone di un *debugger* interno che ho utilizzato per verificare e testare i vari metodi realizzati.

jQuery

jQuery è una libreria *JavaScript* per applicazioni *web*, distribuita come software libero, distribuito sotto i termini della licenza MIT. Ho utilizzato la libreria principalmente per la validazione dei campi all'interno del *form* dell'applicazione *web*, sfruttando il *plugin jQuery Validation*.

In questo modo, ho potuto effettuare una validazione lato *client* dei tipi di dato inseriti dagli utenti, verificandone la coerenza.

Bootstrap

Bootstrap è un *framework front-end* gratuito e *open source* per la progettazione di siti e applicazioni *web*. Contiene modelli di progettazione basati su HTML e CSS per moduli, pulsanti, navigazione e altri componenti dell'interfaccia, nonché estensioni *JavaScript* facoltative.

Per realizzare l'interfaccia dell'applicazione *web* ho sfruttato il tema gratuito e *open source* *SB admin 2*, messo a disposizione dal *framework*.

SQL Server

SQL Server è un *DBMS (Database Management System)* relazionale sviluppato da *Microsoft*. Viene utilizzato per gestire database delle dimensioni e strutture più disparate.

Per interagire e manipolare i dati in *SQL Server*, ho utilizzato *SQL Server Management Studio*, un programma sviluppato appositamente da *Microsoft* il cui utilizzo è già adottato e consolidato in azienda.

Git

Il sistema di *versioning* adottato per questo progetto e in generale utilizzato dall'azienda è Git.

L'azienda adotta attualmente GitHub, un servizio di hosting per progetti basato su Git. Ho utilizzato GitKraken, un client *GUI (Graphic User Interface)* per rendere più facile e veloce l'utilizzo del sistema di versionamento.

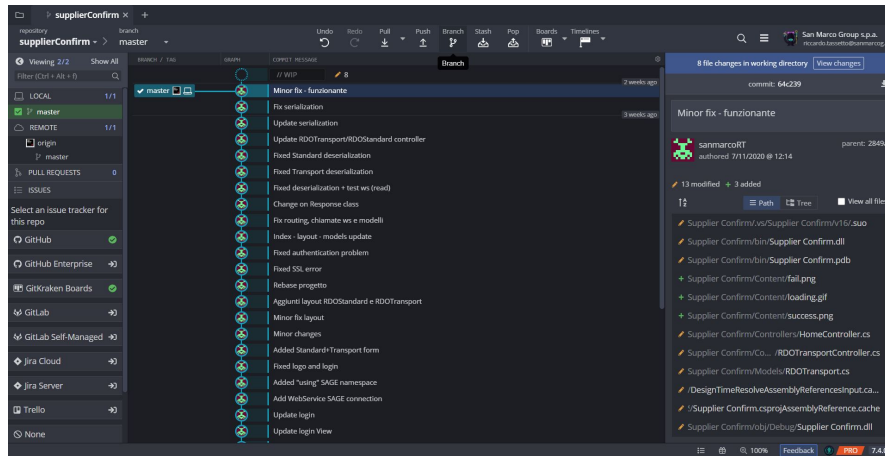


Figura 3.2: Immagine di cattura di schermo che mostra il repository del progetto su Gitkraken.

Tecnologia SOAP

SOAP (simple object access protocol) è un protocollo che regola lo scambio di messaggi tra componenti *software*. Opera principalmente sul protocollo di rete *HTTP*, si basa sul metalinguaggio *XML* e ha una struttura costituita da:

- **header**, che contiene le informazioni necessarie per l'instradamento e la sicurezza;
- **body**, che contiene i dati formattati secondo l'*XML Schema*.

3.2.2 Interfaccia del gestionale

Una volta apprese le nozioni basilari attraverso la documentazione disponibile e tramite un affiancamento con i fornitori del gestionale per conoscere la funzionalità già presente, ho sviluppato l'interfaccia che verrà utilizzata dagli utenti per la gestione delle richieste di offerta.

Introduzione a Sage X3

Sage X3 è un *Enterprise Resource Planning (ERP)*_[G] sviluppato utilizzando un linguaggio proprietario.

La gestione degli oggetti *X3* è alla base della maggior parte delle funzioni messe a disposizione dal *software*.

Un oggetto *X3* (figura 3.3) corrisponde alla gestione completa delle schede di una tabella o di un gruppo di tabelle del database (creazione, consultazione, modifica e annullamento). Ad esempio, la funzionalità di gestione delle richieste di offerta è stata implementata nel *software* gestionale utilizzando un oggetto *X3*, denominato *PQH*.

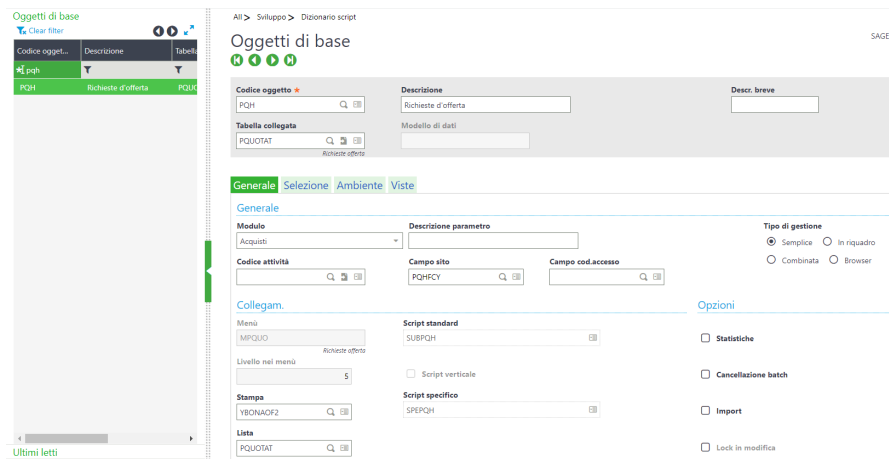


Figura 3.3: Immagine di cattura di schermo che mostra la rappresentazione di un oggetto *X3*.

L'oggetto *PQH* (come possiamo vedere dalla figura 3.4) si presenta come una finestra composta da una testata, da più blocchi (ovvero campi che contengono riferimenti ad altri oggetti *X3*), dai campi che corrispondono ai suoi attributi e da una lista di selezione situata a sinistra chiamata *browser*. Il *browser* permette una ricerca parametrizzata di una specifica istanza di quell'oggetto. Il menù presente sulla destra invece mostra le operazioni messe a disposizione dalla funzionalità, suddivise per *folder*, ovvero dei raggruppamenti di operazioni affini.

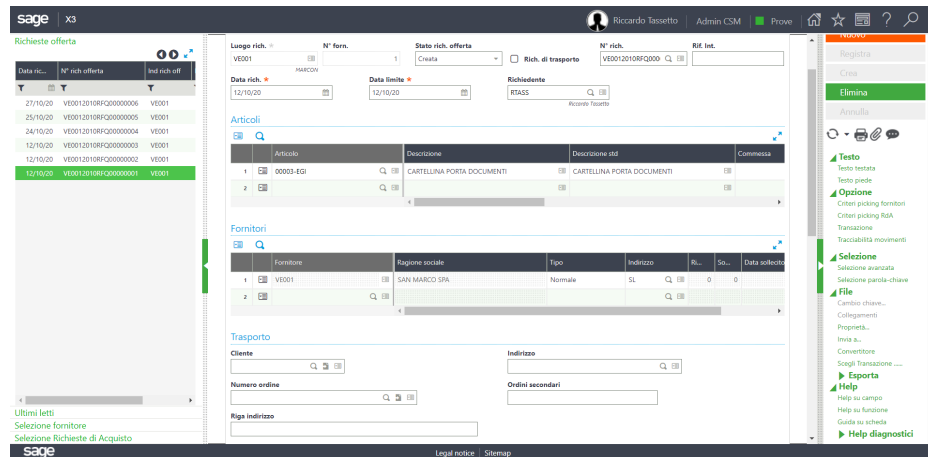


Figura 3.4: Immagine di cattura di schermo che mostra la rappresentazione di una richiesta di offerta.

Personalizzazione dell'oggetto X3

Attraverso la documentazione presente, ho individuato quali tabelle del database erano state utilizzate per generare l'oggetto *X3* che gestiva le richieste di offerta. A quel punto ho potuto, tramite il dizionario delle tabelle presente nell'interfaccia *web*, aggiungere i campi necessari alla gestione delle richieste, individuati durante l'analisi con i *key users*. Per poter riuscire a gestire i due tipi di richieste di offerta (richiesta di beni materiali e richiesta di trasporto) tramite un unico oggetto *X3* ho sfruttato una funzione chiamata **transazione di inserimento**. Questa permette di gestire le parametrizzazioni per la personalizzazione delle videate di inserimento delle richieste d'offerta. Viene inizializzata all'installazione del *software* una transazione *standard* di inserimento per ogni oggetto *X3*. Nel mio caso ho agito in questo modo:

- ho utilizzato la transazione standard (*STD*) per personalizzare la videata di inserimento della richiesta di beni materiali;
- ho creato una nuova transazione (*ALL*) e l'ho utilizzata per personalizzare la videata di inserimento della richiesta di trasporto.

In questo modo, ogni volta che si accede alla funzionalità delle richieste di offerta, compare la scelta delle transazioni e permette di gestire l'uno o l'altro tipo di richiesta, a seconda della necessità.

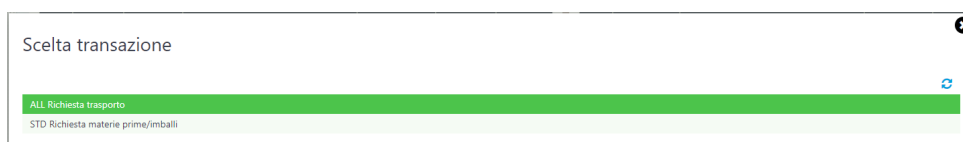


Figura 3.5: Immagine di cattura di schermo che mostra la scelta della transazione all'apertura delle richieste di offerta.

Progettazione dell'interfaccia

L'interfaccia all'interno di *Sage X3* è stata progettata con il *designer web* del gestionale (figura 3.6).

Una volta aggiunti i campi necessari all'oggetto, è stato possibile personalizzare l'interfaccia attraverso:

- *drag and drop* degli elementi presenti a video;
- selezione di un elemento a video e utilizzo di una delle operazioni disponibili sul menù di destra, che operano sulla formattazione e visibilità degli elementi.

L'impostazione data all'interfaccia è stata resa il più possibile simile al modulo cartaceo utilizzato, dividendola in sezioni corrispondenti ai vari uffici competenti.

Funzionamento

Alla creazione di una richiesta di offerta, sia essa di acquisto beni materiali o di trasporto, all'interno di una delle tabelle del database che definisce l'oggetto *X3* viene creato automaticamente un id chiamato *AUUID* che associa univocamente ogni fornitore coinvolto con la relativa richiesta di offerta. Una volta compilata nei relativi

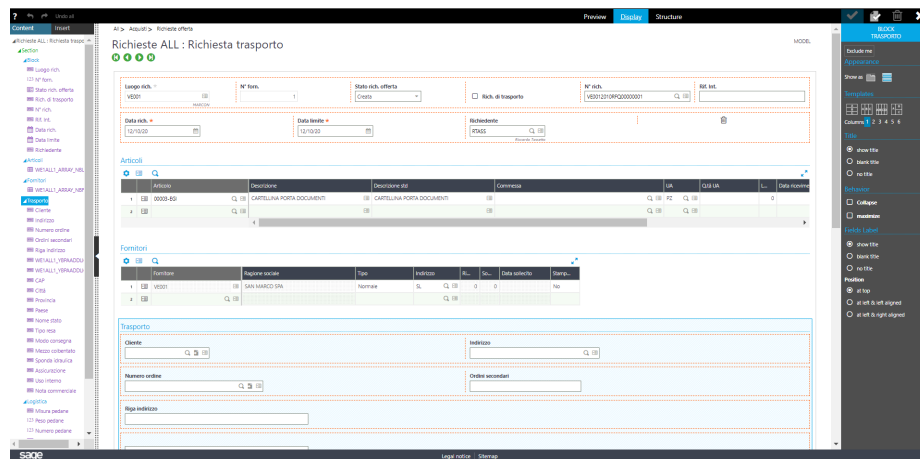


Figura 3.6: Immagine di cattura di schermo che mostra il *designer web* di X3.

campi dagli utenti coinvolti, attraverso la funzionalità presente in *Sage X3* di invio *e-mail*, viene spedita una *e-mail* ad ogni fornitore contenente il corrispondente *AUUIID*, che utilizzeranno come *token* al momento dell'autenticazione nell'applicazione *web*.

3.2.3 Web Service

Dopo la formazione sulla tecnologia non ancora affrontata, ho creato i servizi necessari alla comunicazione bidirezionale tra applicazione *web* e gestionale.

Web service in X3

L'applicazione *web* comunica con la base di dati del gestionale tramite *web service SOAP*. Un *web service SOAP* offre una lista di metodi che possono essere invocati da applicazioni client per svolgere una determinata funzionalità. Ogni *web service* è legato ad un *dossier X3*.

Un *dossier* (di cui vediamo l'architettura in figura 3.7) è una singola ed univoca installazione *X3* che ha a disposizione:

- una piattaforma tecnologica (*runtime server*, *database server*, *web server* ecc.);
- un numero di *folder* con organizzazione gerarchica;
- *directories* sull'*application server*.

Un *dossier* è identificato da un nome e un codice definiti sulla *console web*. Lo stesso *web service* può essere definito su più *dossier*.

All'interno di *Sage X3* è presente una *utility* che permette la pubblicazione di un *web service*: una semplice funzione che trasforma un sottoprogramma o un oggetto *X3* in un *web service* con un nome pubblico. Nel mio caso, ho utilizzato questa *utility* per pubblicare due *web service*, che fornissero i metodi di lettura, modifica e scrittura dell'oggetto

corrispondente alla richiesta di offerta, rispettivamente:

- 1 *web service*, chiamato *WSYPQHSTD*, che offre i metodi per la richiesta di offerta di beni materiali (generata dalla transazione *STD*);

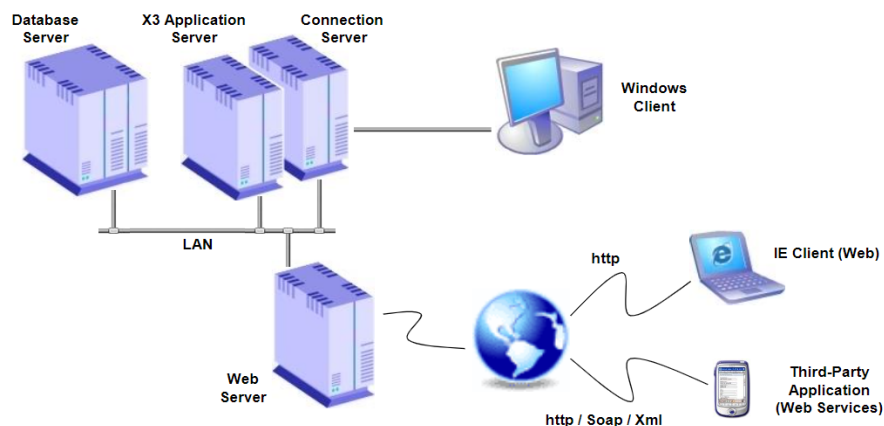


Figura 3.7: Rappresentazione grafica dell'architettura di una installazione di *Sage X3*.

Fonte: Documentazione Online Help Center di *Sage X3*

- 1 *web service*, chiamato *WSYPQHALL*, che offre i metodi per la richiesta di offerta di trasporto (generata dalla transazione *ALL*).

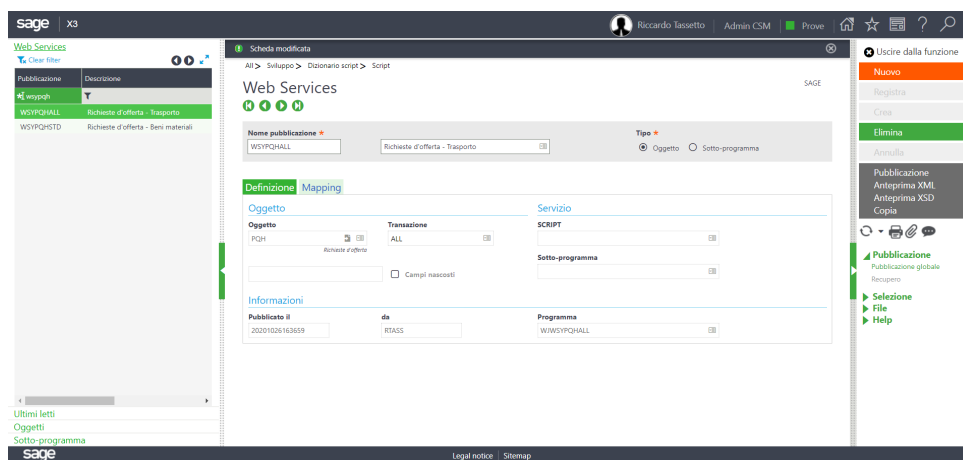


Figura 3.8: Immagine di cattura di schermo che mostra i *web service* creati.

Oltre a questi, ho creato un ulteriore *web service*, chiamato *WSYPQHAUTH*, personalizzando una routine già presente nel *software*. Questo servizio viene utilizzato in fase di autenticazione del fornitore all'interno dell'applicazione *web*. Permette infatti, dato il *token* chiamato *AUUID* inserito dal fornitore, di recuperare il codice univoco della richiesta di offerta a cui è legato, permettendo quindi all'applicazione *web* di recuperare i dati dell'offerta tramite i *web service* *WSYPQHSTD* e *WSYPQHALL* descritti in precedenza.

Gestione della risposta

Per poter gestire le risposte ricevute dai fornitori, ho creato un *web service* sull'oggetto *X3* di risposta alle richieste di offerta (*PPD* chiamato *WSYPPD*). Permette infatti, una volta che il fornitore invia la sua risposta tramite l'applicazione *web*, di creare una nuova istanza dell'oggetto *PPD* contenente i dati inseriti nella risposta. Oltre a questo, l'oggetto contiene anche un riferimento alla richiesta di offerta a cui è associato, in questo modo il *software* gestionale riesce in autonomia a collegare i due oggetti e permettere all'utente aziendale di visualizzarla.

3.2.4 Applicazione web

Il design pattern utilizzato per la realizzazione dell'applicazione *web* è l'*MVC*. Questa tecnica di programmazione separa l'applicazione in tre parti:

- **Model**, che fornisce i metodi per accedere ai dati;
- **View**, che visualizza i dati contenuti nel *model* e permette l'interazione all'utente;
- **Controller**, che si occupa di ricevere gli input dall'utente (attraverso la *view*) e li attua agendo sui dati contenuti nel *model*.

Connessione ad X3

Per poter utilizzare i *web service SOAP* nella mia applicazione *web*, è stato necessario creare una *web reference*, ovvero un riferimento ad un insieme di servizi *web*. Grazie al *tool* messo a disposizione da Visual Studio, ho potuto aggiungere il *WSDL* di *Sage X3*, ovvero un documento *XML* che descrive come interagire con il servizio in questione, che contiene:

- le operazioni che il servizio mette a disposizione;
- il protocollo di comunicazione da utilizzare per accedere al servizio;
- il formato dei messaggi accettati in input;
- il formato dei messaggi restituiti e il loro output;
- gli *endpoint* di ogni funzione.

A questo punto, con i metodi ereditati dal *WSDL*, è possibile implementare una funzione di autenticazione, che andrà successivamente aggiunta all'interno dell'*header* della richieste *SOAP*.

Funzionamento

Il fornitore si collega all'applicazione *web*, dove inserisce nella schermata di autenticazione (figura 3.9) il *token* ricevuto via *e-mail*. Ad autenticazione riuscita, viene caricato il *form* della richiesta (figura 3.10) a lui inviata, permettendogli di vedere in dettaglio le informazioni necessarie.

Se dovesse scegliere di inviare una risposta, dopo aver compilato i campi necessari ha la possibilità di inviarla cliccando sul tasto conferma; a questo punto viene creato un oggetto di risposta *PPD* contenente i dati inseriti nel *form* e il riferimento alla richiesta di offerta associata.

Se invece il fornitore intende rifiutare la richiesta, tramite il pulsante di rifiuto viene inviata una comunicazione che notifica l'utente aziendale di riferimento.

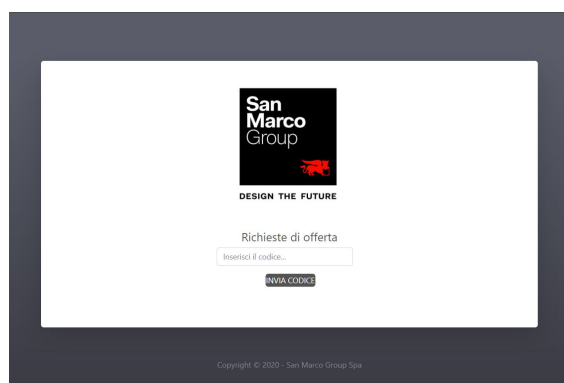


Figura 3.9: Immagine di cattura di schermo del *form* di login dell'applicazione *web*.

The image shows a web browser window displaying a form titled "RICHIESTA DI OFFERTA - TRASPORTO". The form is divided into two columns. The left column contains the following fields: "Data Scadenza" (24/10/2020), "Indirizzo di Presa" (Via Alta 10, MARCON (VE)), "Indirizzo di Consegna" (Via Alta 10, MARCON (VE)), "Resa" (ASS), "Sponda Idraulica" (No), "Assicurazione" (Si), "Misure pedane" (2,50 x 2,00 m), "Prezzo" (150,00 €), and "Data di Consegna" (30/10/2020). The right column contains the following fields: "Nome di Presa" (SAN MARCO GROUP SPA), "Nome di Consegna" (SAN MARCO GROUP SPA), "Tipo di Trasporto" (Corriere espresso), "Mezzo Colbentato" (No), "Articolo" (BLOCCO+PENNA (GREEN SPIRIT)), "Numero pedane" (2), "Peso pedane" (15), "Lead Time" (2 GG), and "Validità Offerta" (30 GG). Below the form fields, there is a "Note" section with a text area containing "Richiesta conferma cliente". At the bottom of the form, there are two buttons: "INVIA OFFERTA" (green) and "RIFIUTA OFFERTA" (red). At the bottom of the page, there is a small copyright notice: "Copyright © 2020 - San Marco Group Spa".

Figura 3.10: *Form* di inserimento risposta nell'applicazione *web*.

3.2.5 Reportistica

Dopo la formazione sullo strumento per la produzione di stampe *Crystal Report*, ho creato il dettaglio della richiesta di offerta. Attraverso il *tool* disponibile sullo strumento, è possibile collegarsi all'origine dati scelta. Ho sfruttato il *template* disponibile in *Sage X3* della bolla di richiesta offerta e personalizzato il *report* seguendo il modello dell'attuale cartaceo utilizzato. Per poter inserire i valori dei campi non ancora presenti sul *report*, è stato sufficiente selezionarli dalle tabelle del database collegate.

Report BI

Dopo un'analisi con il *Purchase Manager* e con il supporto del *tutor* aziendale, ho identificato dei dati interessanti al fine di migliorare il processo. La soluzione proposta permette di tracciare la richiesta dalla sua creazione fino alla risposta dei fornitori. In questo modo è possibile raccogliere una serie di informazioni utili all'ufficio Acquisti che, con le numerose operazioni svolte extra sistema nella precedente gestione, erano difficilmente tracciabili:

- **numero delle richieste di trasporto/numero delle richieste accettate dal cliente**, diventa essenziale individuare quante delle richieste di trasporto che partono dall'ufficio Acquisti e superano la gara dei fornitori, vengono effettivamente accettate dal cliente che richiede il trasporto;
- **numero dei fornitori che rispondono alla richiesta**, altro dato che può essere utilizzato nel metodo di valutazione dell'affidabilità di un fornitore;
- **numero delle richieste di trasporto ad uso interno**, molte richieste vengono fatte con lo scopo di avere una quotazione per un servizio di trasporto e non vengono proposte al cliente finale. Grazie al campo selezionabile all'interno dell'interfaccia utente in *Sage X3* al momento della creazione, è possibile conoscere il numero di queste richieste che andranno considerate diversamente dalle richieste con fine di vendita di un servizio;
- **data proposta/data bolla**, confrontando la data proposta di consegna del materiale acquistato con la data della bolla del trasportatore, che corrisponde quindi all'effettiva consegna della merce, si ha modo di utilizzare questo dato nella valutazione dell'affidabilità del fornitore.

3.3 Verifica e validazione

3.3.1 Verifica

La verifica del *software* è un processo necessario a stabilire che il prodotto e le parti che lo compongono, risponda ai requisiti fissati in precedenza. In un primo momento ho effettuato l'analisi statica, successivamente ho eseguito *test* di unità e integrazione sugli elementi fondamentali del prodotto.

Analisi statica

L'analisi statica è una tipologia di analisi del *software* che non prevede l'esecuzione del codice. Ho effettuato l'analisi statica più volte durante il periodo di *stage*, tramite la lettura del codice prodotto nello sviluppo dell'applicazione *web*, al raggiungimento di ogni punto significativo in termini di codifica.

Test di unità

Per quanto riguarda l'applicazione *web*, attraverso il *framework* *MSTest* presente in Visual Studio sono stati eseguiti i *test* sulle singole unità dell'applicazione *web*, classi e sui metodi principali che compongono il progetto.

Nell'interfaccia del gestionale invece, sono stati verificato il funzionamento dei campi, in particolare sui tipi di dato accettati e il funzionamento delle operazioni di creazione, modifica ed eliminazione delle richieste di offerta.

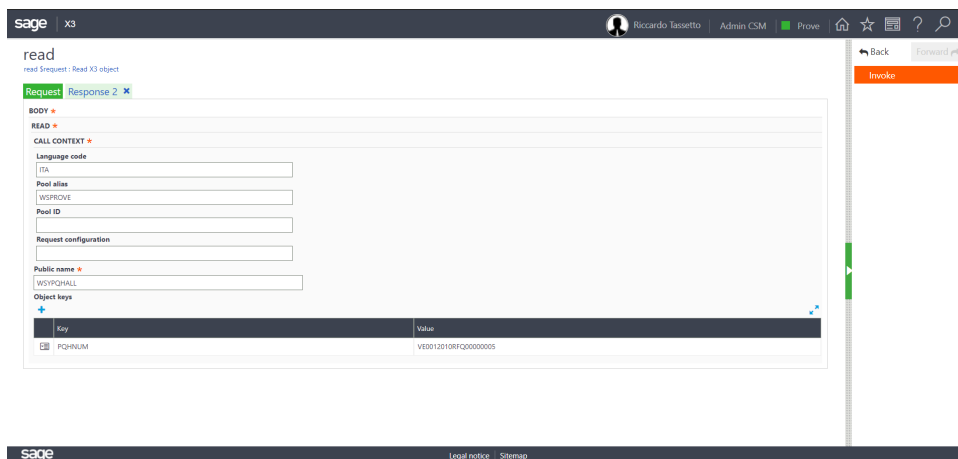


Figura 3.11: Immagine di cattura di schermo che mostra una richiesta dal *web service tester* di *Sage X3*.

Test di integrazione

Essendo ancora in corso lo sviluppo del gestionale, non è stato possibile effettuare *test* sotto stress o che verificassero il funzionamento di funzionalità come l'invio automatico del *token* al fornitore. Sono stati quindi fatti dei *test* ad hoc per verificare il funzionamento delle funzionalità.

Nonostante questo, dopo una valutazione con il *tutor* aziendale siamo arrivati alla conclusione che il lavoro fatto risulta comunque sufficiente.

I *web service* sono stati testati attraverso il *tester* di *Sage X3*. Questo *tool* permette, inserendo i parametri di configurazione, la verifica dei metodi principali disponibili (lettura, modifica, salvataggio, ecc.) sui *web service* pubblicati.

Dalla figura 3.11 vediamo un esempio di *test*. In questo caso viene fatta una richiesta richiamando la funzione *read*, e vengono inseriti nei relativi campi i parametri necessari alla configurazione, come il *dossier* ed il *web service* di riferimento.

La risposta, come possiamo vedere dalla figura 3.12 è costituita da un file *XML* contenente i *tag* dei campi relativi agli attributi della richiesta e il loro valore.

In questo caso il *test* è stato eseguito con successo.

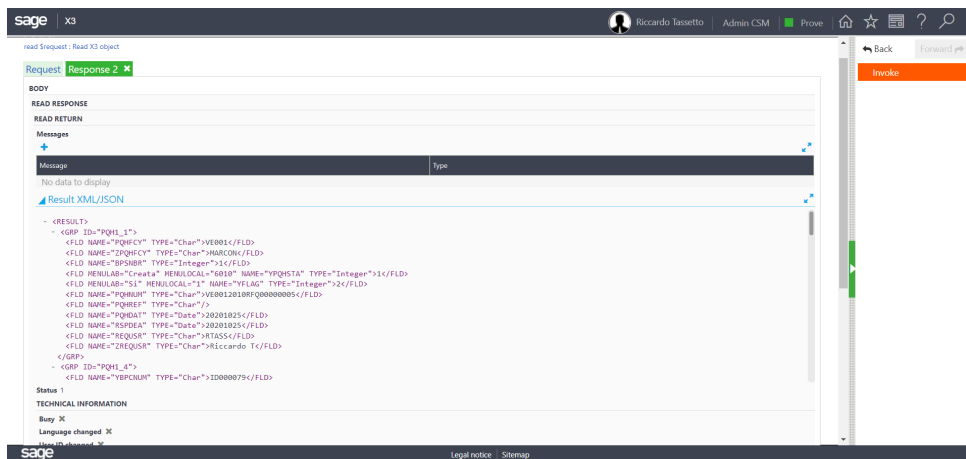


Figura 3.12: Immagine di cattura di schermo che mostra una risposta dal *web service tester* di *Sage X3*.

3.3.2 Validazione

Il processo di validazione del *software* è un controllo che permette di accertare se il prodotto finale corrisponde alle attese e risulta quindi conforme ai requisiti dichiarati inizialmente. Al termine dello *stage* ho effettuato autonomamente una validazione interna verificando il funzionamento dell'interfaccia sviluppata in *Sage X3* e svolgendo i *test* sulle funzionalità implementate nell'applicazione *web*, simulando rispettivamente l'uso da parte di un utente aziendale e l'uso da parte di un fornitore. Essendo ancora in corso lo sviluppo del nuovo gestionale e di tutte le funzionalità previste, ho mostrato il prodotto ed il suo funzionamento al *tutor* aziendale che ha valutato positivamente il lavoro.

3.4 Consuntivo finale

3.4.1 Copertura di requisiti e test

Tutti i requisiti delineati durante la fase di analisi sia relativi all'applicazione *web* che all'interfaccia del gestionale, sono stati soddisfatti.

Tabella 3.3: Tabella di riepilogo requisiti soddisfatti.

| Tipo | Soddisfatti | Non soddisfatti |
|----------------------|--------------------|------------------------|
| Funzionale | 26 | 0 |
| Qualitativo | 1 | 0 |
| Vincolo | 3 | 0 |
| Prestazionale | 0 | 0 |
| Totale | 30 | 0 |

Per quanto concerne la copertura dei *test*, non era stata richiesta una soglia minima di *Code Coverage*. Per l'applicazione *web*, i *test* effettuati fatti sui metodi principali raggiungono il 70% di *Code Coverage*, risultato considerato sufficiente dal *tutor* aziendale. I *web service*, come descritto in 3.3.1, sono stati testati attraverso il modulo integrato in *Sage X3*.

3.4.2 Prodotti realizzati

Come mostrato in figura 3.13, la soluzione realizzata è costituita da tre elementi principali:

- l'interfaccia in *Sage X3*;
- l'applicazione *web*;
- i *web service SOAP*.

L'interfaccia in *Sage X3*, ottenuta personalizzando una funzionalità già nativa nel gestionale, permette agli utenti dell'azienda di rimanere in un unico ambiente di lavoro e gestire le richieste di offerta.

L'applicazione *web* invece, destinata ad essere utilizzata dai fornitori di materie prime e servizi, permette a quest'ultimi di inserire le loro proposte di offerta in risposta alle richieste ricevute.

Infine i *web service SOAP*, che permettono la comunicazione tra l'applicazione e il gestionale, tramite il recupero delle richieste dal database da visualizzare nell'applicazione e la scrittura nello stesso delle offerte ricevute.

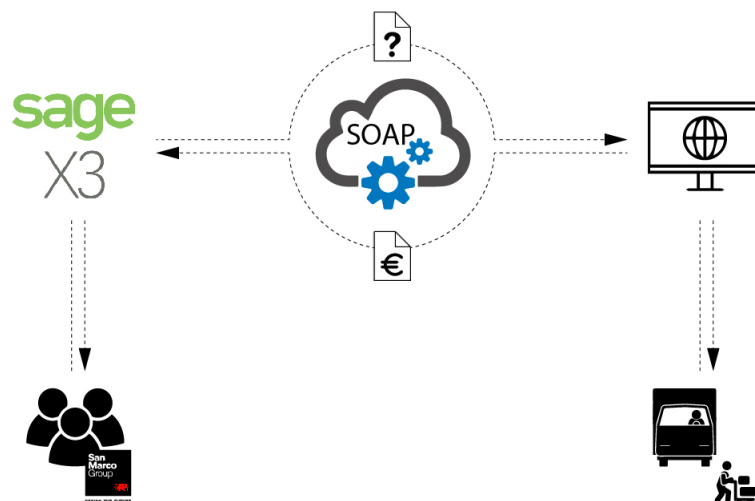


Figura 3.13: Architettura della soluzione realizzata.

Invece, per quanto riguarda i prodotti di tipo documentale richiesti dall'azienda, ho realizzato:

- relazione sul processo aziendale coinvolto;
- relazione sulla progettazione architettonica;
- manuale e documentazione riguardante la struttura dell'applicazione *web* per la manutenzione ed eventuali integrazioni;
- *report* di dettaglio e di analisi *BI*.

Glossario

Business Intelligence

Con la locuzione *Business Intelligence* (*BI*) ci si può solitamente riferire a tre elementi, quali: un insieme di processi aziendali per raccogliere dati ed analizzare informazioni strategiche; la tecnologia utilizzata per realizzare questi processi; le informazioni ottenute come risultato di questi processi. [35](#)

Enterprise Resource Planning

Letteralmente tradotto "pianificazione delle risorse d'impresa", è un *software* di gestione che integra tutti i processi di *business* rilevanti di un'azienda e tutte le funzioni aziendali, ad esempio vendite, acquisti, gestione magazzino, finanza, contabilità, ecc.. Integra quindi tutte le attività aziendali in un unico sistema, il quale risulta essere indispensabile per supportare il *Management*. I dati vengono raccolti in maniera centralizzata nonostante provengano da molteplici parti dell'azienda. [35](#)

ICT

le tecnologie dell'informazione e della comunicazione, acronimo ICT, sono l'insieme dei metodi e delle tecniche utilizzate nella trasmissione, ricezione ed elaborazione di dati e informazioni. [35](#)

Virtual Private Network

Acronimo di *Virtual Private Network*, è un servizio che garantisce *privacy*, anonimato e sicurezza dei dati attraverso un canale di comunicazione riservato tra dispositivi che non necessariamente devono essere collegati alla stessa *LAN*. In ambito prettamente aziendale, una *VPN* può essere paragonata ad una estensione geografica della rete locale privata (*LAN*) e che, quindi, permette di collegare tra loro, in maniera sicura, i siti della stessa azienda dislocati sul territorio. [6](#), [35](#)

Acronimi

BI [Business Intelligence](#). 2, 33

ERP [Enterprise Resource Planning](#). 21, 33

ICT [Information and Communications Technology](#). 15, 33

VPN [Virtual Private Network](#). 6, 33