

Università degli Studi di Modena e Reggio Emilia

Facoltà di Ingegneria di Modena

Corso di Laurea Specialistica in Ingegneria Informatica

**Progetto e realizzazione di
un'applicazione per il controllo dello
sviluppo software in Sistema Qualità**

Relatore:
Chiar.ma Prof.ssa Sonia Bergamaschi

Candidato:
Mauro Verardi

Anno Accademico 2006/2007

Sommario

1. INTRODUZIONE	1
2. SISTEMA QUALITÀ DI UN'AZIENDA DI SERVIZI: IL CASO SYNAGO . 3	3
2.1 QUALITÀ ISO.....	3
2.2 QUALITÀ DEL SOFTWARE	4
2.3 GESTIONE PER LA QUALITÀ E NORME	7
2.4 REQUISITI ISO 9001:2000	9
2.5 MANUALE QUALITÀ SYNAGO.....	14
2.5.1 <i>Processi relativi al cliente</i>	14
2.5.2 <i>Progettazione e sviluppo</i>	15
2.5.2.1 <i>Processo di sviluppo</i>	16
2.5.2.2 <i>Modifiche al piano di sviluppo</i>	17
2.5.2.3 <i>Gestione semplificata del piano di sviluppo</i>	18
2.6 REQUISITI SYNDEV	19
2.6.1 <i>Commessa</i>	20
2.6.2 <i>Fasi e sottofasi</i>	21
2.6.3 <i>Offerta</i>	23
2.6.4 <i>Fatturazione</i>	24
2.6.5 <i>Consolidato</i>	24
2.6.6 <i>Stati commessa-offerta e interazioni</i>	25
2.6.7 <i>Integrazione</i>	27
2.6.8 <i>Cartellino</i>	28
2.6.9 <i>Gantt chart</i>	28
2.6.10 <i>Utenza</i>	30
2.6.11 <i>Altri requisiti</i>	30
2.6.11.1 <i>Tabelle</i>	30
2.6.11.2 <i>Requisiti d'interfaccia utente</i>	32
3. CONTROLLO DI GESTIONE DI UN'AZIENDA DI SERVIZI: IL CASO SYNAGO	34
3.1 PROGRAMMAZIONE E CONTROLLO	34
3.2 LA FIGURA DEL CONTROLLER.....	35

3.3	GLI STRUMENTI DEL CONTROLLO DI GESTIONE	36
3.4	GLI STUDI PROFESSIONALI	40
3.5	REQUISITI SYNDEV	42
3.5.1	<i>Lista commesse</i>	43
3.5.2	<i>Dettagli commessa</i>	45
3.5.3	<i>Riepilogo commesse interne</i>	45
3.5.4	<i>Riepilogo commesse esterne</i>	47
3.5.5	<i>Riepilogo commesse indirette</i>	48
3.5.6	<i>Riepilogo totale</i>	49
3.5.7	<i>Riepilogo risorse</i>	50
3.5.8	<i>Dettaglio risorsa</i>	52
3.5.9	<i>Altri report ed export</i>	53
3.5.10	<i>Casistica commesse</i>	53
4.	IL PROGETTO DI SYNDEV	55
4.1	USE CASE DIAGRAM	55
4.1.1	<i>Gerarchia attori</i>	55
4.1.2	<i>Accesso</i>	56
4.1.3	<i>Cartellino</i>	57
4.1.4	<i>Produzione e pianificazione</i>	58
4.1.5	<i>Amministrazione</i>	59
4.1.6	<i>Controllo</i>	60
4.1.7	<i>Export</i>	60
4.2	ACTIVITY DIAGRAM	61
4.2.1	<i>Accesso al sistema</i>	61
4.2.2	<i>Cartellino</i>	62
4.2.3	<i>Produzione e pianificazione</i>	63
4.2.4	<i>Nuova commessa</i>	64
4.2.5	<i>Pianificazione</i>	65
4.2.6	<i>Amministrazione</i>	66
4.2.7	<i>Controllo – lista commesse</i>	67
4.3	CLASS DIAGRAM	69
4.3.1	<i>Packages</i>	69
4.3.2	<i>Package commesse</i>	70

4.3.3	<i>Package amministrazione</i>	71
4.3.4	<i>Package tabelle</i>	72
4.3.5	<i>Interazioni tra packages</i>	73
4.4	SEQUENCE DIAGRAM.....	75
4.4.1	<i>Accesso</i>	75
4.4.2	<i>Cartellino</i>	76
4.4.3	<i>Produzione</i>	77
4.4.4	<i>Amministrazione</i>	80
4.4.5	<i>Controllo</i>	84
4.5	PROGETTO CONCETTUALE E LOGICO DELLA BASE DI DATI.....	92
4.5.1	<i>Traduzione in schema relazionale</i>	93
4.5.2	<i>Trigger</i>	95
5.	SVILUPPO DI SYNDEV	96
5.1	INTRANET E APPLICAZIONI WEB	96
5.2	FRAMEWORK DI SVILUPPO	98
5.3	ARCHITETTURA DI <i>SYNDEV</i>	99
5.4	DATABASE SYNDEV	102
6.	CONCLUSIONI E SVILUPPI FUTURI	104
	APPENDICE A: ESEMPI REPORT DI CONTROLLO	106
	APPENDICE B: CODICE T-SQL PER LA CREAZIONE DELLE TABELLE E TRIGGER	111
	RIFERIMENTI BIBLIOGRAFICI	120

Indice delle figure

Figura 1: Gestione per la qualità.....	10
Figura 2: Dinamica commessa-offerta	19
Figura 3: Simbologia diagrammi di stato	25
Figura 4: Stati offerta.....	26
Figura 5: Stati commessa.....	26
Figura 6: Gantt commesse	28
Figura 7: Gantt risorse	29
Figura 8: Il controller nell'organigramma aziendale.....	35
Figura 9: Aree strategiche	40
Figura 10: Fasi di gestione di un progetto	41
Figura 11: Semaforo	43
Figura 12: Esame commesse di pertinenza dell'anno in corso	54
Figura 13: Use case – Gerarchia attori	55
Figura 14: Use case – accesso	56
Figura 15: Use case – cartellino	57
Figura 16: Use case – produzione e pianificazione	58
Figura 17: Use case – supporto alla pianificazione	58
Figura 18: Use case – amministrazione.....	59
Figura 19: Use case – Controllo	60
Figura 20: Use case – Export.....	60
Figura 21: Activity diagrams – accesso.....	61
Figura 22: Activity diagrams – cartellino.....	62
Figura 23: Activity diagrams – produzione e pianificazione	63
Figura 24: Activity diagram – nuova commessa	64
Figura 25: Activity diagram – pianificazione.....	65
Figura 26: Activity diagram – amministrazione.....	66
Figura 27: Activity diagram – lista commesse	67
Figura 28: Activity diagrams – riepiloghi	68
Figura 29: Class diagram – packages	69
Figura 30: Class diagram – package commesse	70
Figura 31: Class diagram – package amministrazione	71

Figura 32: Class diagram – package tabelle, attributi	72
Figura 33: Class diagram – interazioni commesse/tabelle	73
Figura 34: Class diagram – interazioni commesse/amministrazione	74
Figura 35: Class diagram – interazioni amministrazione/tabelle	74
Figura 36: Sequence diagram – accesso	75
Figura 37: Sequence diagram – cartellino, risorsa	76
Figura 38: Sequence diagram – cartellino, amministratore	76
Figura 39: Sequence diagram – produzione, lista commesse	77
Figura 40: Sequence diagram – produzione, dettagli commessa	78
Figura 41: Sequence diagram – produzione, nuova commessa	79
Figura 42: Sequence diagram – produzione, nuova fase e nuova sottofase	79
Figura 43: Sequence diagram – amministrazione, lista commesse	80
Figura 44: Sequence diagram – amministrazione, dettagli commessa	81
Figura 45: Sequence diagram – amministrazione, visualizza consolidato	82
Figura 46: Sequence diagram – amministrazione, crea integrazione	83
Figura 47: Sequence diagram – controllo, lista commesse	84
Figura 48: Sequence diagram – controllo, dettagli commessa	85
Figura 49: Sequence diagram – controllo, riepilogo commesse interne	86
Figura 50: Sequence diagram – controllo, riepilogo commesse esterne	87
Figura 51: Sequence diagram – controllo, riepilogo commesse indirette	88
Figura 52: Sequence diagram – controllo, riepilogo commesse totale	89
Figura 53: Sequence diagram – controllo, riepilogo risorse	90
Figura 54: Sequence diagram – controllo, dettaglio risorsa	91
Figura 55: Schema E/R	92
Figura 56: Trigger	95
Figura 57: Architettura di Syndev	101
Figura 58: Controllo - lista commesse	106
Figura 59: Appendice A, controllo - dettagli commessa	106
Figura 60: Appendice A, controllo - riepilogo commesse interne	107
Figura 61: Appendice A, controllo - riepilogo commesse esterne	107
Figura 62: Appendice A, controllo - riepilogo commesse indirette	108
Figura 63: Appendice A, controllo - riepilogo totale	109
Figura 64: Appendice A, controllo - riepilogo risorse	109
Figura 65: Appendice A, controllo - dettaglio risorsa	110

1. Introduzione

La presente tesi è stata svolta durante un periodo di stage della durata di 5 mesi presso l'azienda *Synago S.r.l.* di Parma.

Il progetto proposto dall'azienda ospitante ha avuto come obiettivo la progettazione e lo sviluppo di un software per la gestione di alcuni aspetti dell'andamento aziendale, la precedente gestione avveniva con i programmi *Syndev* e *Cartellino*, che sono stati oggetto di analisi per il nuovo sistema. Per lo sviluppo di tale progetto si è quindi partiti dai programmi già utilizzati in azienda per lo svolgimento delle stesse funzioni ma implementati solo in forma prototipale. Durante l'esposizione il nome "Syndev" verrà usato per riferirsi all'applicazione oggetto della tesi. Si possono riconoscere tre macrocategorie alle quali imputare le funzionalità richieste:

- **Produzione:**

gestione dei dettagli inerenti i processi produttivi che si svolgono in azienda, pianificazione e monitoraggio delle risorse aziendali;

- **Amministrazione:**

gestione delle offerte rivolte ai clienti e delle fatturazioni associate ad ogni commessa;

- **Controllo:**

elaborazioni ed aggregazioni dei dati al fine di generare report sull'andamento generale dell'azienda per favorire il processo decisionale della direzione aziendale.

Synago S.r.l. opera attraverso un *Sistema qualità* certificato *UNI EN ISO 9001: 2000*, il software che si vuole realizzare si propone quindi come strumento a sostegno del corretto svolgimento delle attività aziendali nell'ambito di tale sistema.

Parte del periodo di stage è stato impiegato per la raccolta dei requisiti, per chiarire le problematiche da affrontare e determinare le funzioni che il software andrà a svolgere state svolte interviste e discussioni con i futuri utenti dell'applicazione, si è studiato il *Manuale di qualità* aziendale ed analizzato i programmi di gestione usati in azienda. Successivamente si è passati alla fase di progettazione elaborando un documento secondo lo standard UML. Poi si è scelto un paradigma architetturale ed un framework considerando le varie alternative esistenti e le possibilità aziendali.

La tesi è organizzata nel seguente modo:

Nel **capitolo 1** vengono presentate le problematiche inerenti i sistemi di gestione per la qualità in azienda facendo riferimento al Manuale Qualità di *Synago S.r.l.* e alle relative norme *ISO*, sono approfonditi gli aspetti riguardanti i processi produttivi e viene spiegato in che modo il software progettato si inserisce in tale sistema.

Il **capitolo 2** descrive la funzione del controllo di gestione, gli obiettivi, le tecniche e i vantaggi per un'azienda che adotta un sistema di controllo di gestione. Viene poi illustrato come si è affrontato il problema nell'ambito di questo progetto.

Alla fine dei **capitoli 1 e 2** si troverà l'esposizione dei requisiti che *Syndev* dovrà soddisfare relativamente ai problemi affrontati in ognuno di essi.

Nel **capitolo 3** viene illustrato il progetto del software descritto usando lo standard UML come linguaggio di modellazione, lo schema concettuale della base di dati secondo il modello Entity-Relationship e lo schema relazionale risultante.

Il **capitolo 4** espone la fase di sviluppo dell'applicazione descrivendo la piattaforma utilizzata, l'architettura adottata e la traduzione dello schema relazionale della base di dati progettata nell'estensione SQL del DBMS impiegato.

2. Sistema Qualità di un'azienda di servizi: il caso Synago

Synago S.r.l. ha optato per l'adozione di un *sistema qualità* certificato ISO 9001:2000 che regola gli aspetti operativi della gestione aziendale. In questo capitolo verranno affrontate le problematiche relative la *gestione di qualità* e verrà spiegato come il lavoro svolto durante la presente tesi si inserisce nel *sistema qualità* dell'azienda ospitante.

Per definire cosa sia un *sistema di qualità* e la *qualità* stessa ci si può riferire alle normative emesse dall'organizzazione *ISO*, acronimo di *International Organization for Standardization*. Tale organizzazione non governativa si occupa della preparazione e pubblicazione di Standard internazionali, ovvero svolge attività per stabilire, relativamente a problemi effettivi o potenziali, disposizioni per gli usi comuni e ripetitivi, miranti ad ottenere l'ordine migliore in un determinato contesto^[1]. *ISO* è in realtà formato da membri di organismi di normazione di 157 paesi diversi provenienti da ambiti sia pubblici che privati, infatti alcuni di essi sono mandati direttamente dal proprio governo o da strutture governative mentre altri dal settore privato o associazioni industriali.

2.1 Qualità ISO

Una prima spiegazione della *qualità* si trova già nel documento *ISO 8402:1995* che fissa i termini e le definizioni fondamentali relativi alla *qualità*, ai *sistemi qualità* e agli strumenti e alle tecniche per tenerla sotto controllo, è definita come

“L'insieme delle caratteristiche di un'entità che ne determinano le capacità di soddisfare esigenze espresse ed implicite”^[2].

Nella più recente norma *ISO 9000:2000* tale definizione viene aggiornata e si identifica la qualità con

“il grado con cui un insieme di caratteristiche intrinseche soddisfa i requisiti”^[3],

dove per *requisito* si intende

“un’esigenza o aspettativa che può essere espressa,
generalmente implicita o cogente”^[3].

Si potrebbe pensare che con il termine requisito ci si riferisce alle caratteristiche di un prodotto, ciò è vero solo in parte in quanto il concetto è più ampio, si può estendere infatti anche in altri ambiti, esistono ad esempio requisiti di *gestione della qualità* (v. *2.4 Requisiti ISO 9001:2000*) piuttosto che requisiti del cliente.

2.2 Qualità del software

A titolo esemplificativo si illustra in questo paragrafo la normativa in materia di qualità del software. Con il progetto *SQuaRE*, acronimo di *Software Quality Requirements and Evaluation*, *ISO* vuole armonizzare le tante norme già emesse in materia di qualità del software e pubblica quindi le guide *ISO 250xx*. La nuova famiglia di norme *ISO 25000* può essere suddivisa in 5 sezioni contenenti^[4]:

- termini e definizioni per la qualità del prodotto software e la sua valutazione;
- un modello di riferimento della qualità del software;
- un modello di riferimento per la misura e valutazione del software;
- un modello di riferimento per la definizione dei requisiti del software;
- un modello di riferimento per la pianificazione e gestione della qualità del software.

In particolare le principali pubblicazioni a cui riferirsi sono^[5]:

- **ISO/IEC 2500n** definiscono i modelli e termini generali comuni al progetto *SQuaRE* e forniscono una guida e una serie di suggerimenti pratici agli altri documenti *SQuaRE*;
- **ISO/IEC 2501n** presentano un modello di qualità che descrive caratteristiche interne, esterne ed in uso (v. sotto);
- **ISO/IEC 2502n** descrivono un modello di riferimento per la misura della qualità di un prodotto software in riferimento alle caratteristiche interne, esterne ed in uso (v. sotto);

- **ISO/IEC 2503n** sono un supporto per la definizione di requisiti di qualità per lo sviluppo di un prodotto software o per il processo di valutazione;
- **ISO/IEC 2504n** forniscono requisiti, raccomandazioni e linee guida per la valutazione di un prodotto software.

In *ISO/IEC 25000* sono fissate tre categorie di *caratteristiche di qualità* del software^[5]:

- **esterne:** proprietà dinamiche del software rilevabili quando viene usato, si collocano quindi a lato utente. Ad esempio appartengono a questa categoria:
 - *correttezza:* un sistema software è corretto se si comporta come è previsto nella specifica dei requisiti;
 - *affidabilità:* un sistema è tanto più affidabile quanto più raramente si presentano malfunzionamenti;
 - *robustezza:* è la misura in cui il sistema si a in modo accettabile a fronte di situazioni impreviste;
 - *efficienza:* proprietà legata al consumo di risorse;
 - *usabilità:* è una caratteristica soggettiva in quanto un sistema è facile da usare se l'utente lo reputa tale, dipende quindi dall'esperienza e dal contesto;

- *scalabilità*: capacità del sistema di crescere o decrescere in funzione della necessità e disponibilità. Tipicamente ci si riferisce alla scalabilità di carico.
 - *fault-tolerance*: capacità del sistema di non subire interruzioni di servizio anche a seguito di guasti;
- **interne**: proprietà statiche del software riferite ad architettura e struttura relative alla qualità percepita dagli sviluppatori. Ad esempio si collocano in questa categoria:
- *verificabilità*: grado di facilità con cui è possibile controllare le proprietà di correttezza e affidabilità;
 - *manutenibilità*: facilità di apportare modifiche al software realizzato, sia per correzione di errori sia per miglioramento;
 - *riparabilità*: difficoltà di correzione di errori;
 - *evolubilità*: caratteristica legata alla facilità con cui il software può evolversi nel futuro, è necessario quindi tener conto degli sviluppi futuri già nella fase di progettazione;
 - *riusabilità*: in che misura le parti del sistema software possono essere riutilizzate in un altro progetto, ci si può riferire non solo al prodotto finale ma anche ad altri fattori, come le specifiche e i progetti;
 - *portabilità*: in sistema è portabile se può essere utilizzato in ambienti diversi;

- **d'uso:** proprietà legate al contesto d'uso dell'utente:
 - *efficacia:* proprietà legata alla completezza informativa, all'aggiornamento delle informazioni;
 - *produttività:* caratteristica legata alla rilevanza e comprensibilità dei dati;
 - *safety:* capacità dei dati di garantire un livello accettabile di rischio di danno a persone o business;
 - *soddisfazione:* credibilità dei dati e accessibilità.

2.3 Gestione per la qualità e Norme

Una volta chiarito cosa si intende per *qualità* si può definire *gestione per la qualità* come l'insieme di

“attività coordinate per guidare e tenere sotto controllo un'organizzazione in materia di qualità”^[3].

Di regola, per guidare e tenere sotto controllo un'organizzazione in materia di *qualità*, si definiscono

- politica ed obiettivi per la *qualità*;
- pianificazione della *qualità*;
- controllo della *qualità*, definita come:

“parte della *gestione per la qualità* mirata a soddisfare i requisiti per la qualità”^[3];

nella norma *ISO 8402* il termine *controllo della qualità* era invece definito come

"Tecniche ed attività a carattere operativo messe in atto per soddisfare i requisiti per la qualità"^[2];

in entrambi i casi si comprende il carattere operativo del *controllo della qualità* che consente di stabilire la conformità di un prodotto o di un servizio ai requisiti specificati. Il *controllo* si può riferire quindi ad esempio alla progettazione di componenti, alla loro esecuzione e installazione o a prodotti software per i quali valgono norme specifiche (v. 2.2 *Qualità del software*) o anche allo stesso sistema di gestione.

- assicurazione della *qualità*:

“parte della *gestione per la qualità* mirata a dare fiducia che i requisiti per la qualità saranno soddisfatti”^[3];

- miglioramento della *qualità*:

“parte della *gestione per la qualità* mirata ad accrescere la capacità di soddisfare i requisiti per la qualità”^[3].

Un *sistema di qualità* ha la caratteristica principale di essere orientato al cliente, cioè ha come scopo l'accrescimento della soddisfazione del cliente.

Le norme in vigore in materia di *sistema di gestione* appartengono alla famiglia *ISO 9000*, ad essa appartengono:

- **ISO 9000:2000** descrive i concetti e termini fondamentali nell'ambito della famiglia;
- **ISO 9001:2000** specifica i requisiti di un modello di *sistema di gestione per la qualità* per le organizzazioni indipendentemente dal tipo, prodotti e servizi forniti o dimensioni;

- **ISO 9004:2000** contiene le linee guida per adempiere ai requisiti descritti nella *ISO 9001:2000* e per un successivo miglioramento delle prestazioni di un'organizzazione.

2.4 Requisiti ISO 9001:2000

La norma *ISO 9001:2000* contiene i requisiti che un *sistema qualità* deve soddisfare per godere della certificazione.

Approccio per processi

Come prima direttiva la norma promuove l'adozione all'interno dell'organizzazione di un *approccio per processi* da applicare al *sistema di gestione per la qualità*^[6]. Un *processo* è ogni attività che utilizza delle risorse al fine di trasformare degli elementi in ingresso in elementi in uscita. Un'organizzazione deve quindi identificare tali processi e determinarne le interazioni in quanto l'output di un processo può essere l'input di un secondo processo. Un approccio del genere implica l'importanza di comprendere e osservare i requisiti del prodotto e di un continuo miglioramento dei processi stessi a seguito di valutazioni basate su misurazioni oggettive. A tal proposito si può applicare ad ogni processo il modello PDCA, o ciclo di Deming:

- **Plan:** stabilire gli obiettivi ed i processi necessari per fornire risultati in accordo con i requisiti del cliente e le politiche dell'organizzazione;
- **Do:** attuare i processi;
- **Check:** monitorare e misurare processi;
- **Act:** attuare misure di miglioramento.

Nella normativa viene inoltre illustrato il modello di *gestione della qualità basato sui processi* come si vede in *Figura 1*.

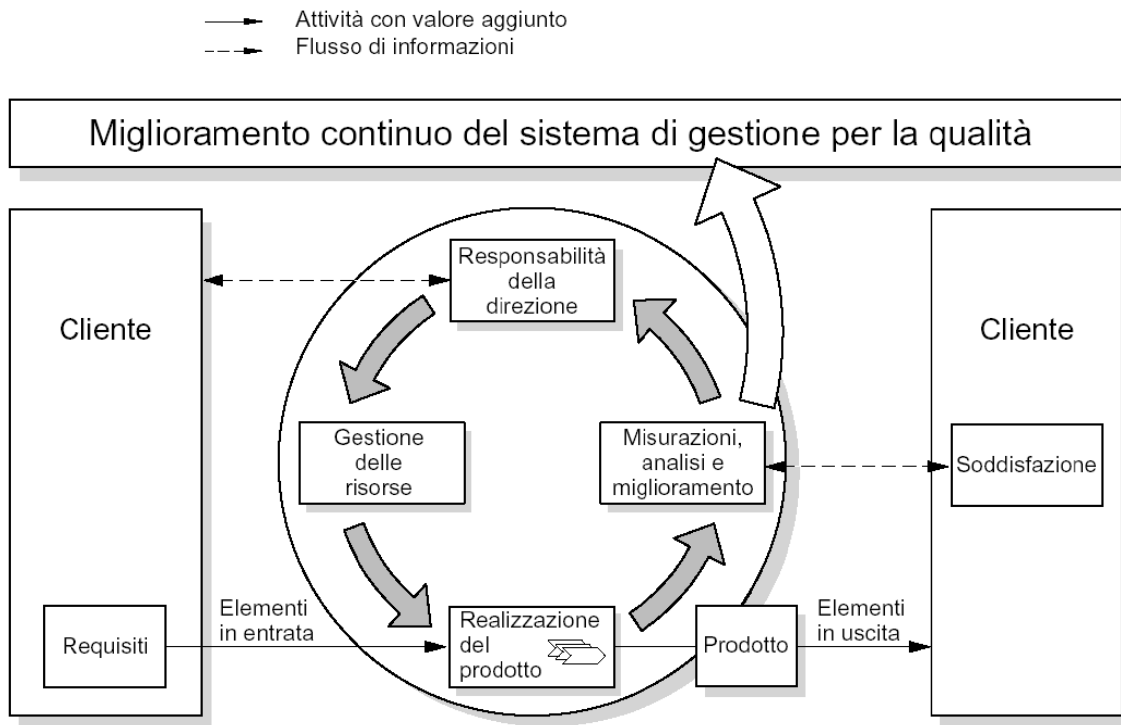


Figura 1: Gestione per la qualità

Si può notare come sia gli elementi in entrata, i requisiti, che in uscita, il prodotto, provengono dal cliente nel primo caso e pervengono al cliente nel secondo. Ci sono anche altre interazioni tra l'organizzazione e il cliente con lo scopo di capire quali sono gli interventi necessari da attuare sui processi e perseguire la soddisfazione del cliente. Tutto ciò è in accordo con la politica di orientamento al cliente di un *sistema di qualità*.

Riguardo l'*approccio per processi* la norma impone di^[7]:

- identificare i processi per la gestione per la qualità;
- stabilire sequenza e interazione tra di essi;
- stabilire criteri e metodi per assicurare l'efficace funzionamento e controllo di tali processi;
- monitorare, misurare e analizzare i processi;
- attuare azioni correttive per il perseguimento dei risultati pianificati e il continuo miglioramento.

Gestione delle risorse

In ambito di gestione delle risorse la norma dispone che l'organizzazione debba individuare e rendere disponibili le risorse necessarie per accrescere la soddisfazione del cliente soddisfacendo i requisiti e per attuare il sistema di *gestione per la qualità* tenendolo aggiornato e migliorandone l'efficacia^[8]. Per quanto riguarda le risorse umane l'organizzazione deve occuparsi dell'addestramento del personale e renderlo consapevole della rilevanza delle sue azioni, deve inoltre conservare registrazioni sulla competenza e l'addestramento del personale .

Sempre al fine di ottenere la conformità dei prodotti ai requisiti previsti l'organizzazione deve fornire un ambiente di lavoro adatto e le necessarie infrastrutture quali gli stessi spazi di lavoro, le attrezzature di processo (hardware o software) e servizi di supporto (ed esempio trasporto e comunicazione).

Documentazioni

La norma impone all'organizzazione certificata di mantenere determinati documenti riguardo il *sistema di gestione per la qualità*^[9], vale a dire una dichiarazione riguardo le politiche e gli obiettivi per la qualità, un *manuale della qualità*, tutti i documenti necessari per assicurare l'efficace pianificazione, funzionamento e controllo dei processi e le procedure documentate nonché le registrazioni richieste direttamente nella norma.

Successivamente si farà riferimento al *manuale di qualità*^[9] di *Synago S.r.l.* per motivare alcuni dei requisiti funzionali di *Syndev* in quanto, oltre alle informazioni riguardo il campo di applicazione ed esclusioni giustificate del *sistema per la qualità*, esso deve descrivere le procedure documentate predisposte per il *sistema di gestione per la qualità* e le interazioni tra tali processi.

Realizzazione del prodotto

Un intero paragrafo della norma è dedicato alla spiegazione dei requisiti dei processi che regolano la realizzazione del prodotto, in particolare compaiono le seguenti sezioni^[10]:

- pianificazione della realizzazione del prodotto;
- processi relativi al cliente;
- progettazione e sviluppo;
- approvvigionamento;
- produzione ed erogazione di servizi;
- tenuta sotto controllo dei dispositivi di monitoraggio e di misurazione.

La sezione relativa alla pianificazione prevede^[13] che l'organizzazione definisca gli obiettivi per la qualità ed i requisiti relativi al prodotto, i processi e i documenti relativi alla produzione, le registrazioni che evidenzino la soddisfazione dei requisiti e le attività di verifica, validazione, monitoraggio, ispezione, prove specifiche e criteri di accettazione per il prodotto.

Relativamente al rapporto tra organizzazione e cliente la norma prevede^[14] che quest'ultima determini i requisiti per il prodotto specificati dal cliente unitamente a quelli non specificati ma necessari, vanno considerati inoltre i requisiti cogenti e quelli stabiliti dall'organizzazione stessa. La gestione della realizzazione del prodotto deve anche prevedere il riesame dei requisiti prima dell'impegno da parte dell'organizzazione di fornire il prodotto stesso. La norma inoltre prevede che si fissino delle modalità efficaci di comunicazione con il cliente in merito alle informazioni riguardanti il prodotto, il contratto, gli ordini, i quesiti e i feedback dal cliente stesso.

Riguardo alla progettazione e sviluppo la norma impone^[15] all'organizzazione di pianificare le fasi, le attività di riesame, di verifica e validazione e di fissare le responsabilità e le autorità per la progettazione e lo sviluppo. Inoltre si pronuncia riguardo gli elementi in ingresso alla progettazione e sviluppo, essi devono essere i requisiti funzionali, prestazionali e cogenti e altre informazioni attinenti. La norma

descrive anche le caratteristiche che gli elementi in uscita devono soddisfare, prevede il riesame, la verifica, la validazione e il monitoraggio della progettazione e sviluppo.

Nel caso dell'azienda ospitante l'attività di approvvigionamento è modesta e saltuaria^[11] in quanto, essendo un'azienda di servizi informatici non necessita di un costante rifornimento di materie prime, inoltre i prodotti o servizi acquistati non anno necessariamente influenza sul prodotto fornito ai clienti.

Relativamente alla produzione ed erogazione di servizi la norma impone^[15] all'organizzazione che le attività vengano pianificate ed avvengano in condizioni controllate, fissa requisiti riguardo l'identificazione e la rintracciabilità del prodotto lungo tutte le fasi della produzione e lo stato di avanzamento del prodotto.

Per quanto riguarda le direttive sui dispositivi di monitoraggio e misurazione Synago srl è esonerata dal seguire tali requisiti in quanto non utilizza simili strumenti^[12].

Altri requisiti

La norma definisce dei requisiti anche riguardo il ruolo e la responsabilità della direzione^[16] riguardo la qualità e affronta le problematiche di misurazione, analisi e miglioramento^[17] per dimostrare e assicurare la conformità dei prodotti e aumentare l'efficacia del *sistema di gestione per la qualità*.

2.5 Manuale qualità Synago

Nel presente paragrafo si farà riferimento al *Manuale di qualità* di Synago S.r.l. (v. 2.4 *Requisiti ISO 9001:2000*) per introdurre i requisiti fissati per *Syndev* durante il periodo dello stage presso l'azienda stessa. In realtà si prenderà in considerazione esclusivamente il capitolo riguardante la realizzazione del prodotto in quanto *Syndev* si propone come strumento informatico per la produzione e l'amministrazione in regime di qualità.

Nel primo paragrafo del capitolo riguardante la realizzazione del prodotto^[18] viene introdotta la pianificazione della modalità di svolgimento delle attività per la realizzazione dei prodotti e servizi e viene ripresa l'organizzazione per processi adottata dall'azienda.

Successivamente vengono illustrati i processi che regolano il rapporto tra l'organizzazione e il cliente^[19], la progettazione e lo sviluppo^[20], l'approvvigionamento^[21], e la produzione ed erogazione dei servizi^[22].

2.5.1 Processi relativi al cliente

Nella premessa del paragrafo^[19] vengono ripresi i principi della norma che regolano i rapporti tra clienti e organizzazione come precedentemente accennato nel paragrafo 2.4 *Requisiti ISO 9001:2000* della presente tesi.

Successivamente^[23] viene illustrato il procedimento previsto per la gestione del contratto, il ciclo di vita tipico dello sviluppo in *Synago S.r.l.* si sviluppa secondo le seguenti fasi:

- **contatto:** si raccolgono le richieste del cliente unitamente a tutte le altre informazioni reperite ed eventuali documenti tecnici;
- **identificazione dei requisiti:** vengono definite le attività necessarie alla gestione dello sviluppo consentendo la fase successiva di valutazione di fattibilità, l'identificazione dei requisiti potrebbe anche coincidere con la fase di

contatto nel caso in cui i documenti e informazioni raccolte in tale fase siano sufficienti^[24];

- **valutazione di fattibilità:** partendo dai documenti prodotti nella fase precedente il responsabile commerciale valuta la fattibilità del lavoro e la definizione dell'offerta economica;
- **redazione offerta:** l'offerta viene redatta dal responsabile commerciale, eventualmente in collaborazione con altri responsabili aziendali e deve essere approvata dalla direzione di essere presentata al cliente previa revisione responsabile commerciale;
- **risposta del cliente:** il cliente può decidere di accettare l'offerta piuttosto che rifiutarla o richiedere un riesame del contratto.

Nel caso venga richiesto un riesame del contratto si ripercorre lo stesso iter sopraillustrato per la redazione di una nuova offerta.

2.5.2 Progettazione e sviluppo

Il capitolo relativo alla progettazione e sviluppo del manuale precisa che “per *sviluppo* si intende l'insieme delle attività che concorrono alla trasformazione dei dati e requisiti in prodotto e/o servizio software attraverso le fasi progettazione e realizzazione^{«[20]}”.

La gestione del processo di sviluppo definisce^[23]:

- come dati e requisiti in input vengono identificati, documentati e riesaminati;
- il ciclo di vita dello sviluppo;
- le fasi della progettazione e sviluppo;
- input e output;
- le attività di riesame, verifica e validazione;
- metodologie e regole di sviluppo;
- documentazione ed esecuzione dei test;

- risorse impiegate e livelli di responsabilità;
- interfacce tra i gruppi coinvolti;
- l'aggiornamento degli elementi in uscita dalla pianificazione con il progredire della progettazione e sviluppo;
- le modalità di presentazione dell'offerta;
- l'identificazione e rintracciabilità del software.

La modalità appena illustrata viene applicata alle attività appartenenti ad entrambe le aree di lavoro dell'azienda ospitante^[23], vale a dire *area Host* e *area Web*, tuttavia la struttura dello sviluppo è flessibile, non esistono cioè vincoli nella sequenzialità delle fasi se non strettamente legati ai requisiti e necessità dell'attività in questione.

2.5.2.1 *Processo di sviluppo*

Il *processo di sviluppo* è suddiviso nelle seguenti fasi^[24]:

- contatto (v. 2.5.1 *Processi relativi al cliente*);
- definizione requisiti base (v. 2.5.1.f.i *Processi relativi al cliente*);
- caratterizzazione commessa;
- pianificazione;
- definizione dati e requisiti;
- progettazione;
- realizzazione;
- rilascio e consegna;
- assistenza;
- controllo.

Una volta che i requisiti di base sono stati identificati viene avviato il sottoprocesso di *caratterizzazione della commessa* che comprende le fasi di^[25]:

- definizione del livello di rischio in termini di costi rischi e complessità;
- definizione di massima delle fasi;

- redazione della scheda di sviluppo;
- riesame e valutazione di fattibilità del progetto;

Successivamente viene decisa una *pianificazione della commessa* che viene approvata e tenuta aggiornata.

Vengono poi definiti con chiarezza e senza ambiguità i requisiti di base raccolti nella fase di *definizione dei requisiti di base*.

Il seguente sottoprocesso di *progettazione* è legato allo sviluppo del software e se necessario può a sua volta svilupparsi in fasi quali la definizione dei livelli di progettazione, dei requisiti funzionale e l'analisi di massima e di dettaglio.

Nel sottoprocesso di *realizzazione* si procede con la scrittura di nuovi programmi o modifica di già esistenti, l'input possono essere i documenti di analisi e progetto nel primo caso oppure i soli dati e requisiti nel secondo caso o se il cliente ha già fornito della documentazione tecnica.

Dopo la realizzazione si procede con *rilascio e consegna* del prodotto ed eventuale *assistenza*.

Il sottoprocesso di *controllo* è come un supervisore degli altri in quanto prevede il monitoraggio della commessa, riesami, validazioni e test per raggiungere la conformità degli output degli altri sottoprocessi.

2.5.2.2 *Modifiche al piano di sviluppo*

Il *sistema di gestione per la qualità* prevede, a seguito di verifiche, validazioni o aggiornamenti dei requisiti da parte del cliente di apportare modifiche al piano di sviluppo^[26]. Nel caso le modifiche da apportare non siano consistenti si può aprire un'anomalia o, al contrario, è necessario iniziare un nuovo *piano di sviluppo*.

2.5.2.3 *Gestione semplificata del piano di sviluppo*

Come già accennato nell'introduzione del presente paragrafo (v. 2.5.2 *Progettazione e sviluppo*) le fasi del processo di sviluppo non sono strettamente sequenziali e necessarie per ogni attività^[27], infatti si possono presentare casi in cui lo sviluppo è talmente semplice che non è necessario stilare un piano di sviluppo completo, ad esempio può succedere che le fasi di *riesame*, *verifica* e *validazione* possono essere accorpate alla fase conclusiva di *rilascio e consegna*.

2.6 Requisiti Syndev

Dalla metodologia illustrata nel paragrafo 2.5.1 *Processi relativi al cliente* emerge che tra le entità che il sistema Syndev dovrà gestire compaiono:

- **commessa:** identifica l'insieme delle attività finalizzate alla soddisfazione delle richieste del cliente;
- **offerta:** è la proposta economica associata ad una commessa.

Si può già delineare la dinamica di queste due entità:

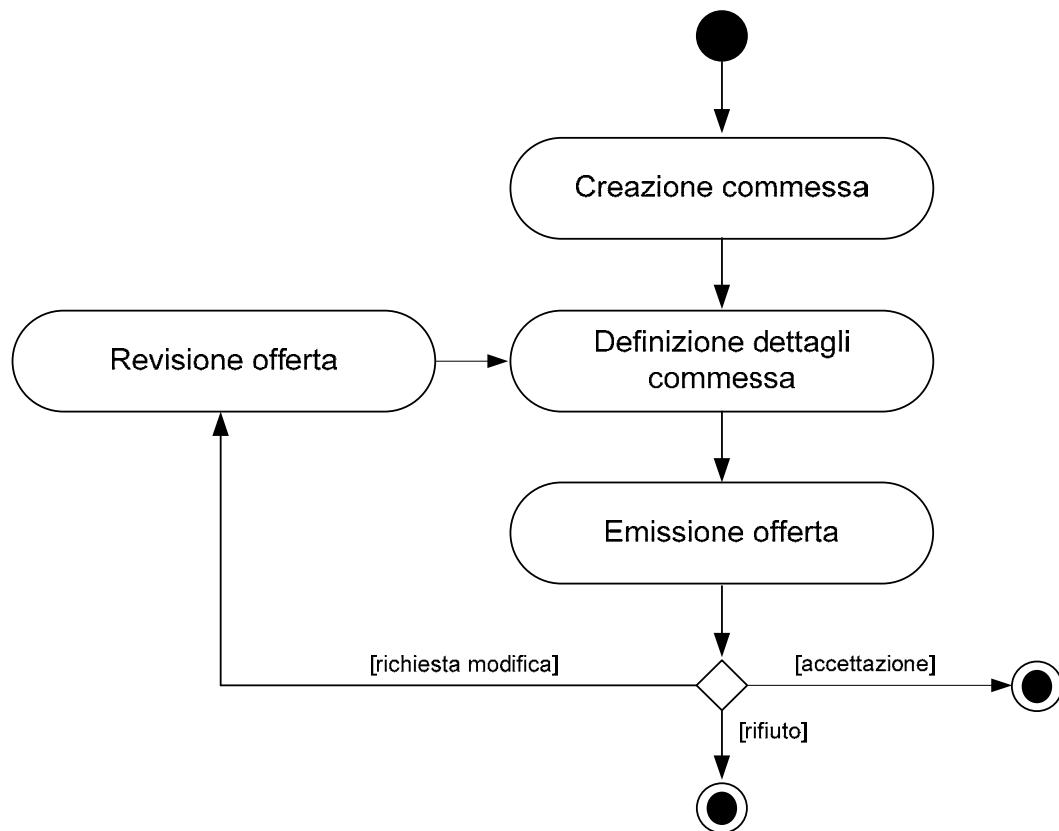


Figura 2: Dinamica commessa-offerta

Dopo che si è creata la commessa si decidono le attività da svolgere per soddisfare le richieste del cliente e si emette l'offerta. La richiesta di modifica dell'offerta porta alla revisione della commessa e all'emissione di una nuova offerta. L'accettazione dell'offerta trasforma la stessa offerta in contratto e permette l'inizio dei lavori per soddisfare le richieste del cliente. Il rifiuto dell'offerta chiude la commessa.

2.6.1 Commessa

Dall'analisi del *manuale di qualità* dell'azienda ospitante unitamente alle interviste e discussioni effettuate nella prima parte del tirocinio aziendale emergono gli attributi delle commesse che Syndev deve gestire:

- **dati identificativi:** informazioni di base per l'identificazione e descrizione della commessa:
 - *codice*;
 - *titolo*;
 - *cliente*;
- **tipo di attività:** appartenenza a una delle aree aziendali *Host* o *Web* (v. 2.5.2 *Progettazione e sviluppo*) e specifica se si tratta di lavoro interno o di assistenza presso il cliente (v. 2.5.2.1 *Processo di sviluppo*):
 - *area*;
 - *ambiente*;
- **responsabilità:** identifica il responsabile delle attività della commessa in questione (v. 2.5.2.1 *Processo di sviluppo*):
 - *responsabile progetto*;
- **stato sviluppo:** la commessa può essere *aperta* se i lavori sono ancora in corso o *chiusa* se tutti i requisiti del cliente sono stati raggiunti o l'offerta non è stata accettata (v. 2.5.1 *Processi relativi al cliente*):
 - *stato sviluppo*;

- **livello di rischio:** definizione dei costi, rischi e complessità:
 - *livello rischio;*
- **pianificazione:** pianificazione dello sviluppo (v. 2.5.2.1 *Processo di sviluppo*):
 - *data inizio commessa;*
 - *data fine commessa;*
 - *definizione fasi* (v. 2.6.2 *Fasi e sottofasi*);
 - *stima percentuale realizzazione;*
 - *data stima;*
- **altro:** informazioni ausiliarie alla caratterizzazione della commessa e il corretto svolgimento delle attività:
 - *note interne private;*
 - *prerequisiti;*
 - *procedura/progetto:* indica l'appartenenza dalla commessa ad un progetto più ampio.

2.6.2 Fasi e sottofasi

Il paragrafo 2.5.2.1 *Processo di sviluppo* elenca le varie fasi del processo di sviluppo, la fase di *contatto* e di *definizione dei requisiti base* non sono strettamente legate alla produzione e quindi non compariranno tra gli elementi gestiti da *SyndeV*.

La fase *caratterizzazione della commessa* consiste nella definizione delle informazioni elencate nel paragrafo precedente e delle fasi come è illustrato in questo paragrafo, prevede quindi l'utilizzo di *SyndeV* stesso e di conseguenza nemmeno questa fase comparirà tra gli elementi gestiti dal sistema.

Le restanti tipologie invece compaiono in *SyndeV* come attributo da associare ad ogni singola fase del *processo di sviluppo*. Quindi per ogni *fase* saranno specificati i seguenti attributi:

- **dati identificativi:** informazioni di base per l'identificazione e descrizione della fase:
 - *codice fase;*

- *nome*;
- *tipo fase*;
- **altro:** informazioni ausiliarie:
 - *note*.

Nel paragrafo 2.5.2 *Progettazione e sviluppo* emerge la necessità di definire le risorse da assegnare allo svolgimento di ogni fase. A tal proposito è stato scelto di assegnare ad ogni fase un numero variabile di *sottofasi* e assegnare una risorsa ad ognuna di esse. Quindi una *sottofase* sarà caratterizzata da:

- **dati identificativi:** informazioni di base per l'identificazione della sottofase:
 - *codice sottofase*;
- **dati risorsa:** identificazione della risorsa associata e del ruolo:
 - *codice risorsa*;
 - *figura professionale*;
- **pianificazione:** pianificazione dettagliata delle attività:
 - *data inizio*;
 - *data fine*;
 - *giorni stimati*;
 - *terminazione* ;
- **altro:** informazioni ausiliarie:
 - *note*.

2.6.3 Offerta

Come emerge da 2.5.1 *Processi relativi al cliente* l'offerta è un documento a carattere economico, è la proposta commerciale fatta al cliente e che potrebbe essere accettata, rifiutata o rettificata. Gli attributi che sono stati identificati per l'*offerta* sono i seguenti:

- **dati identificativi:** informazioni di base per l'identificazione dell'offerta:
 - *codice commessa corrispondente;*
 - *descrizione offerta;*
- **dettagli operativi:** informazioni riguardo attività e persone coinvolte:
 - *dettaglio attività;*
 - *interfacce con il cliente;*
 - *destinatario dell'offerta;*
- **dettagli offerta:** informazioni riguardo giorni di lavoro e somma oggetto dell'offerta:
 - *data emissione;*
 - *giorni/figura professionale;*
 - *importo;*
- **dettagli esito:** informazioni sull'accettazione da parte del cliente:
 - *codice accettazione;*
 - *data accettazione;*
 - *giorni accettati;*
 - *importo accettato;*
- **dettagli commerciali:** informazioni riguardo la fatturazione:
 - *competenza fatturazione;*
 - *fatturazioni* (v. paragrafo successivo);
- **altro:** informazioni aggiuntive:
 - *note.*

2.6.4 Fatturazione

Tramite l'offerta viene concordata con il cliente il numero di rate con cui effettuare il pagamento, vengono quindi associate all'offerta un numero variabile di *fatture previste* caratterizzate da:

- **dati identificativi:**
 - *numero fattura;*
 - *data fattura;*
- **ammontare fattura:**
 - *importo fattura;*
- **altro:**
 - *note.*

SyndeV registrerà anche le *fatture emesse* con le stesse caratteristiche appena elencate per le *fatture previste*, il numero delle due tipologie di fatture non deve essere necessariamente lo stesso per la stessa commessa ma il sistema segnalerà eventuali anomalie e il completamento della fatturazione basandosi sul confronto tra l'importo previsto totale e l'importo fatturato totale.

2.6.5 Consolidato

Quando l'offerta viene accettata il sistema deve registrare l'intero *piano di sviluppo* accettato. Viene quindi creato il *consolidato* della commessa, ovvero la collezione delle seguenti informazioni al momento dell'offerta:

- dettagli commessa consolidata (v. 2.6.1 *Commessa*);
- dati identificativi, dettagli operativi e totale giorni/figura professionale dell'offerta (v. 2.6.3 *Offerta*);
- pianificazione fasi (v. 2.6.2 *Fasi e sottofasi*);

2.6.6 Stati commessa-offerta e interazioni

Tra le commesse, le corrispondenti offerte e fatturazioni vi sono delle forti interazioni riguardo le transazioni di stato delle stesse. Gli stati previsti per le commesse e le offerte sono:

- commessa:
 - inserita;
 - modificare;
 - offerta emessa;
 - offerta accettata;
 - offerta rifiutata;
 - in fatturazione;
 - fatturazione chiusa;
 - chiusa;

- offerta:
 - precompilata;
 - emessa;
 - accettata;
 - rifiutata;
 - modificare.

Le *figure 4 e 5*, utilizzando la simbologia dei diagrammi di stato dello standard UML (v. *Figura 3*), mostrano i cambiamenti di stato di offerta e commessa anche in funzione dell'emissione delle fatture:

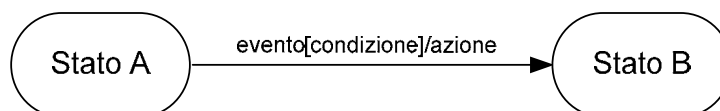


Figura 3: Simbologia diagrammi di stato

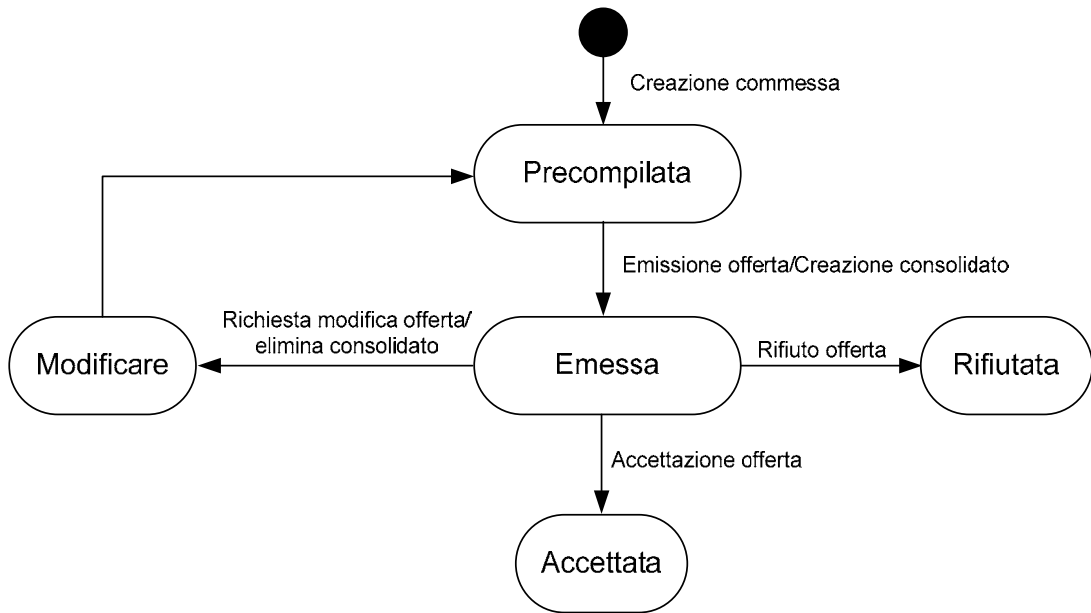


Figura 4: Stati offerta

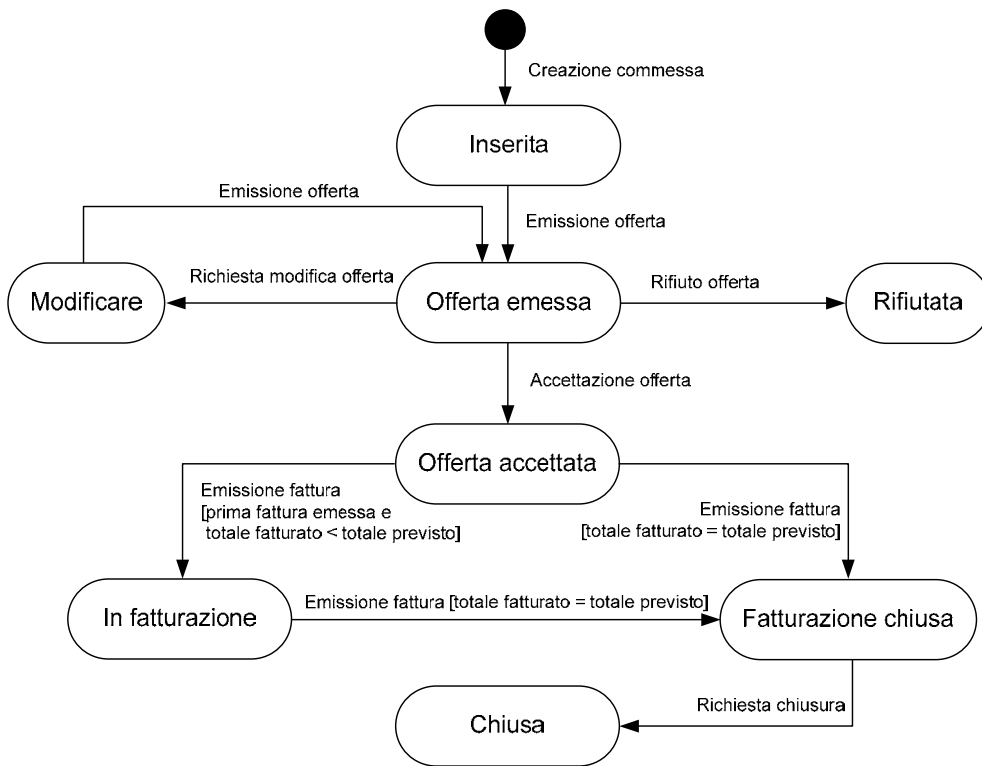


Figura 5: Stati commessa

Le azioni di creazione commessa, può essere effettuata sia dai *responsabili* che dall'*amministratori* mentre le azioni per la gestione dell'offerta sono di esclusiva competenza dell'amministrazione (v. 2.6.10 *Utenza*).

Tener traccia dello stato delle *commesse* e delle *offerte* è importante in quanto ogni stato determina le operazioni che l'utente può svolgere su di esse. Infatti se l'*offerta* è stata emessa non è possibile apportare delle modifiche ne al *piano di sviluppo* ne all'*offerta* stessa, di contro se l'*offerta* è accettata è possibile modificare solo la pianificazione (*Fasi e sottofasi*) per motivi di gestione interna poiché c'è comunque il *consolidato* (v. paragrafo precedente) a memorizzare il *piano di sviluppo* al momento dell'emissione ed eventuale accettazione dell'*offerta* da parte del cliente.

2.6.7 Integrazione

In accordo con quanto descritto nel paragrafo 2.5.2.2 *Modifiche al piano di sviluppo* Syndev fornisce, per i casi in cui è necessario aprire un'anomalia, una funzionalità per la definizione di *integrazioni*. L'*integrazione* è una particolare commessa utile per registrare del lavoro richiesto dal cliente in via straordinaria e già effettuato, infatti lo *stato sviluppo* e necessariamente *terminato*, non c'è la possibilità di specificare un *piano di sviluppo* in termini di *fasi e sottofasi* (v. 2.6.2 *Fasi e sottofasi*) e non c'è il *consolidato* (v. 2.6.5 *Consolidato*), per il resto vanno inserite tutte le informazione come descritto nei paragrafi 2.6.1 *Commessa*, 2.6.3 *Offerta* e 2.6.4 *Fatturazione*.

2.6.8 Cartellino

Per tener traccia dell'avanzamento dei *piani di sviluppo Syndev* fornisce uno strumento di monitoraggio degli stessi con la registrazione dei *cartellini*. Un cartellino è caratterizzato da:

- *identificatore sottofase;*
- *identificatore risorsa corrispondente;*
- *data;*
- *numero ore.*

In altre parole, con la compilazione dei cartellini da parte delle risorse in base alle ore dedicate ad ogni commessa di loro competenza, si può stimare il livello di completamento della commessa a partire dai giorni di lavoro pianificati

2.6.9 Gantt chart

Un altro strumento per ottimizzare l'occupazione delle risorse e massimizzare la disponibilità delle stesse per soddisfare i requisiti del cliente (v. 2.4 *Requisiti ISO 9001:2000*) è la generazione di *diagrammi di Gantt* che descrivano lo stato sia delle commesse che delle risorse nel tempo.

La *figura 6* è un esempio di diagramma di Gantt per le commesse:

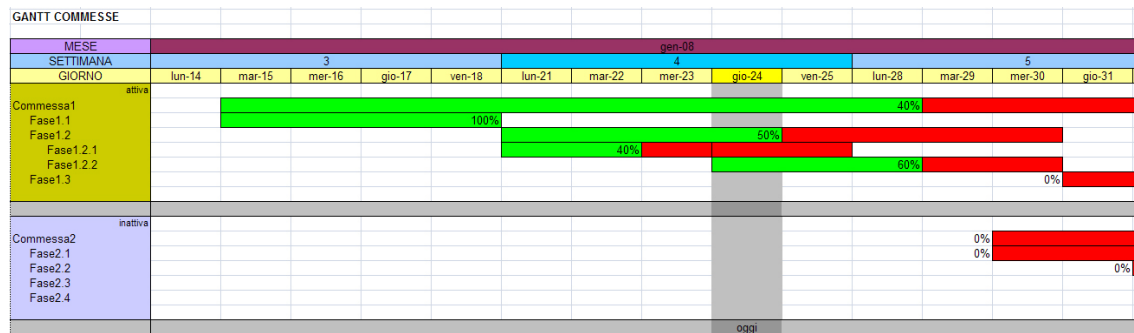


Figura 6: Gantt commesse

Il diagramma presenta le commesse in un determinato intervallo di tempo con la corrispondente percentuale di stima realizzazione. Tale valore può essere impostato dal responsabile progetto oppure calcolato dal sistema secondo la seguente formula:

$$\sum_{\text{sottofase}} \left(\frac{\text{giorni lavorati}}{\text{giorni stimati}} \right)$$

dove i *giorni stimati* sono un attributo proprio della sottofase (v. 2.6.2 *Fasi e sottofasi*) e i *giorni lavorati* sono la somma delle ore nei cartellini della sottofase considerate (v. 2.6.8 *Cartellino*).

La *figura 7* mostra invece un esempio di diagramma di Gantt delle risorse:

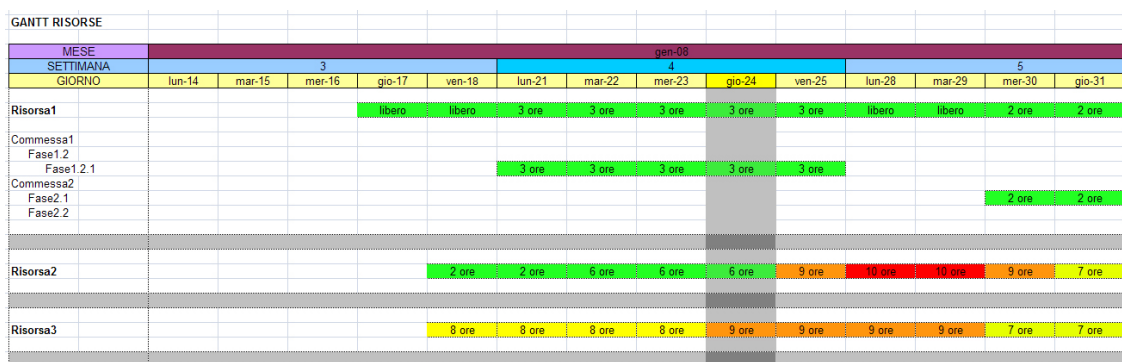


Figura 7: Gantt risorse

Il diagramma presenta l'occupazione totale e relativa ad ogni commessa per ogni singola risorsa in un certo intervallo di tempo. L'occupazione in un giorno è calcolata come segue:

$$\sum_{\text{sottofase}} \left(\frac{\text{giorni rimasti da lavorare}}{\text{giorni stimati}} \right) \times 8 \text{ ore}$$

dove i *giorni rimasti da lavorare* sono calcolati come *giorni stimati* – *giorni lavorati*. Il primo fattore moltiplicativo è una percentuale, quindi opzionalmente si può scegliere di visualizzare la percentuale di occupazione piuttosto che le ore occupate.

Si può notare che è possibile che una risorsa sia occupata più delle normali 8 ore lavorative e si potrebbe pensare a un'anomalia, ma ciò è del tutto normale in quanto si tratta di stime e non di ore effettive di lavoro.

2.6.10 Utenza

SyndeDev prevede tre tipologie di utenti:

- **Risorsa:** corrisponde al dipendente dell'azienda, la sola operazione permessa è la gestione del proprio cartellino (v. 2.6.8 *Cartellino*);
- **Amministratore:** ha la possibilità di accedere a tutte le funzionalità del sistema compresa la possibilità di compilare il cartellino altrui (v. 2.6.8 *Cartellino*);
- **Responsabile:** di progetto o di area sviluppo, gode dei stessi privilegi dell'utente *Amministratore* limitatamente alle risorse di propria pertinenza.

2.6.11 Altri requisiti

2.6.11.1 Tabelle

SyndeDev gestisce anche altre entità che hanno come caratteristica comune la necessità di fornire solo le funzionalità di inserimento, cancellazione e, solo per comodità, di modifica. Tali registrazioni hanno anche la peculiarità di non variare nel tempo se non sporadicamente. Si tratta infatti di informazioni che cambiano con i dettagli del *sistema di gestione per la qualità* stesso. Durante lo svolgimento dell'attività progettuale dello stage è stato assegnato a tali informazioni il nome di "tabelle" e sostanzialmente contengono i valori selezionabili di molti attributi delle altre entità principali descritte in questo capitolo. Le *tabelle* sono le seguenti:

- **ambiente:** informazioni riguardo le aree dell'azienda:
 - *Host*;
 - *Web*;
- **attività:** momento in cui si valutano i costi (v. 2.6 *Requisiti SyndeDev*):
 - *a preventivo*;
 - *a consuntivo*;

- **cliente:** lista clienti dell'azienda;
- **competenza:** anni selezionabili come competenza fatturazione (v. 2.6.3 *Offerta*);
- **costo orario:** costo orario di ogni risorsa (v. 2.6 *Requisiti Synde*), accessibile solo da utenti *amministratori* (v. 2.6.10 *Utenza*);
- **figura professionale:** lista figure professionali selezionabili (v. 2.6.2 *Fasi e sottofasi*):
 - *coordinatore*;
 - *capo progetto*;
 - *analista*;
 - *analista programmatore*;
 - *programmatore*;
 - *programmatore junior*;
- **livello rischio:** lista livelli di rischio selezionabili (v. 2.6.1 *Commessa*):
 - *alto*;
 - *medio*;
 - *basso*;
- **procedura/progetto:** progetto o procedura di appartenenza di una commessa (v. 2.6.1 *Commessa*);
- **risorsa:** lista delle risorse aziendali, poiché coincidono con gli utenti che accedono a Synde nella *tabella risorse* vengono mantenute le informazioni di autenticazione della risorsa al sistema e il tipo di utente, *risorsa semplice* o *amministratore* (v. 2.6.10 *Utenza*).
- **responsabile:** lista dei responsabili, di progetto o di sviluppo, un responsabile corrisponde ad una ed una sola risorsa (v. 2.6.10 *Utenza*);
- **settore attività:** contiene le categorie di attività come elencate in *1.f.ii.1 Processo di sviluppo* con informazioni riguardo documentazioni, input e output;

- **tipo fase:** lista dei tipi di fase selezionabili (v. 2.6.2 *Fasi e sottofasi*), ogni elemento della lista è associato ad uno e un solo *settore attività*.

2.6.11.2 *Requisiti d'interfaccia utente*

Dalle interviste e discussioni con i futuri utenti di *SyndeV* è emersa la preferenza di suddividere le operazioni da svolgere su commesse e offerte in due categorie:

- **produzione:**
- **amministrazione.**

Appartengono alla *produzione* le operazioni di inserimento dati come descritto nei paragrafi 2.6.1 *Commessa* escluso l'anno di competenza, 2.6.2 *Fasi e sottofasi* e 2.6.3 *Offerta* limitatamente ai dettagli operativi, giorni/figura professionale e note.

Le operazioni *amministrative* riguardano l'anno di competenza descritto nel paragrafo 2.6.1 *Commessa*, i dati enumerati nel paragrafo 2.6.3 *Offerta* non appartenenti alla *produzione* (v. sopra) e le informazioni relative alla fatturazione (v. 2.6.4 *Fatturazione*). La parte amministrativa del sistema è accessibile solo dagli utenti *amministratori* (v. 2.6.10 *Utenza*).

È stato inoltre richiesto che le commesse siano accessibili tramite un elenco aggiornabile dinamicamente in base a dei filtri di ricerca diversi tra *produzione* e *amministrazione*:

- **produzione:**
 - *intervallo temporale;*
 - *stato sviluppo;*
 - *area;*
 - *cliente;*
 - *attività;*
 - *ambiente;*
 - *responsabile progetto;*

- *procedura/progetto;*
- **amministrazione:**
 - *intervallo temporale;*
 - *competenza;*
 - *stato commessa;*
 - *stato sviluppo;*
 - *area;*
 - *cliente;*
 - *attività;*
 - *ambiente;*
 - *responsabile progetto;*
 - *procedura/progetto.*

3. Controllo di gestione di un'azienda di servizi: il caso Synago

Il *controllo di gestione* è stato definito in vari modi dagli studiosi: come insieme di strumenti e tecniche quantitativi finalizzati al miglioramento del processo decisionale o come un processo di autocontrollo direzionale (v. 2.5.2.1 *Processo di sviluppo*). Di sicuro si può identificare il fine del *controllo di gestione* nel controllo del conseguimento degli obiettivi programmati e soprattutto verifica che la gestione si svolga secondo criteri di economicità al fine di consentire il perdurare dell'impresa nel tempo^[28], infatti uno degli strumenti del *controllo di gestione* è proprio la contabilità. Ciò che spinge un'azienda ad adottare un *sistema di controllo* è da un lato l'esigenza di affrontare i problemi derivanti dalla complessità interna e dall'altro è un aiuto a reagire con tempestività alle turbolenze dell'ambiente esterno, in realtà, alla luce di questi due obiettivi, si distingue tra *controllo di gestione* e *controllo della strategia*. Si può dire, quindi, che un *sistema di controllo* fornisce un'insieme di informazioni, utili alla direzione, riguardo la qualità di funzionamento dell'azienda.

Per perseguire tali obiettivi, gli elementi fondamentali in un *sistema di controllo* sono^[28]:

- le tecniche per il rilevamento tempestivo dei dati utili alla valutazione;
- gli obiettivi, ovvero i punti di riferimento che determinano il buon andamento della gestione;
- un sistema di reporting che evidenzia chiaramente se e dove avviene un disallineamento tra obiettivi fissati e risultati misurati.

3.1 Programmazione e controllo

La funzione di *controllo di gestione* si posiziona a valle della *programmazione*, intesa come il processo che, sulla base delle strategie aziendali, individua le azioni da eseguire nel breve periodo, tipicamente fino ad un anno, stabilendo le risorse da impiegare e fissando compiti e responsabilità delle diverse aree aziendali^[30], da distinguere dalla *pianificazione strategica* con la quale la direzione stabilisce obiettivi e strategie aziendali e si pone quindi nel lungo periodo. Le attività di *controllo* forniscono

quindi informazioni riguardo i risultati ottenuti che, confrontati con gli obiettivi pianificati, permettono di valutare la validità delle scelte operate e definire più efficacemente le azioni correttive da intraprendere per raggiungere un ulteriore miglioramento (v. modello *PDCA* in 2.4 *Requisiti ISO 9001:2000*). Si può ulteriormente fare una distinzione tra *controllo direzionale* e *controllo operativo*, il primo rappresenta il processo mediante il quale la direzione si assicura che le risorse siano acquisite ed usate efficacemente ed efficientemente per raggiungere gli obiettivi aziendali, la seconda è invece relativa agli specifici compiti^[29].

3.2 La figura del controller

Con *controller* si intende chi si occupa professionalmente del controllo di gestione. Poiché il *controller* ha come obiettivo monitorare le dinamiche interne e fornire supporto nel processo decisionale, si pone tipicamente vicino alla direzione generale:

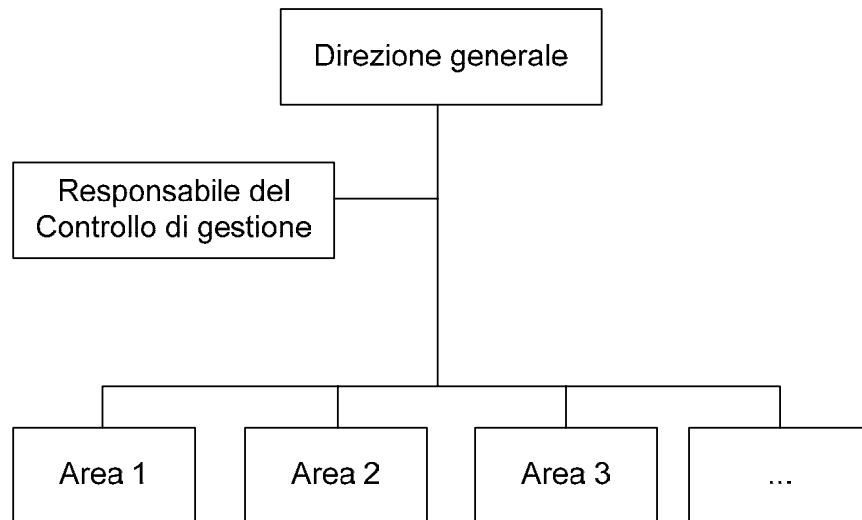


Figura 8: Il controller nell'organigramma aziendale

A seconda della dimensione e articolazione dell'azienda le funzioni di controllo possono essere accorpate ad altre e assegnate ad un unico ufficio o direzione di solito denominato *amministrazione, finanza e controllo* oppure si distingue tra^[31]:

- **finanza:** raccolta e gestione delle disponibilità monetarie e risorse finanziarie;
- **amministrazione:** redazione bilancio e adempimenti fiscali;
- **controllo:** rilevazione interna, budgeting, preparazione e analisi report (v. prossimo paragrafo).

3.3 Gli strumenti del controllo di gestione

Affinché i processi di programmazione e controllo raggiungano gli scopi è necessario prendere in considerazione la tendenza che si ha di decentrare le attività, il potere decisionale e quindi le responsabilità. Ciò significa avere a che fare con una gerarchia che parte dalla direzione generale e si sviluppa attraverso i responsabili delle varie unità con concessione di poteri diversi e conseguenti responsabilità. Per valutare la relazione decisioni – azioni – risultati è quindi necessario^[32]:

- individuare le aree organizzative critiche;
- scegliere i parametri obiettivo e gli indicatori adatti.

Per quanto riguarda il primo punto in genere rispecchia la struttura organizzativa stessa dell'azienda, bisogna poi decidere la granularità degli elementi monitorati dal sistema.

Riguardo i parametri da scegliere è evidente che quelli di tipo economico-finanziari sono molto indicativi ma non è detto che siano i soli ad essere presi in considerazione.

La valutazione di due diverse figure all'interno dell'organizzazione potrebbero essere valutate quindi con parametri e criteri diversi senza escludere che più tipologie di valutazioni vengano effettuate sullo stesso operatore, tipicamente si usa distinguere in base a^[32]:

- **costo:** l'efficienza è legata ai costi sostenuti;

- **spesa:** dato un livello minimo qualitativo dell'operato si assegna un tetto massimo di spesa da non superare e si valuta in base a tale limite;
- **ricavo:** l'efficienza è legata esclusivamente al volume di ricavi da raggiungere;
- **profitto:** si considerano sia i ricavi che i costi, si valuta quindi in base ai risultati operativi;
- **investimento:** fissati i risultati da raggiungere si valutano gli investimenti effettuati.

È chiaro che per raggiungere i propri obiettivi il *controllo di gestione* necessita di informazioni basate su rilevazioni di risultati, soprattutto a livello contabile. Lo strumento che riesce a soddisfare tale esigenza è il *sistema informativo* aziendale, cioè l'insieme delle attività, metodi e strumenti tecnologici per la raccolta, elaborazione e presentazione delle informazioni nell'organizzazione. A seconda delle esigenze tale sistema può essere più o meno articolato e formato da sottosistemi, tra questi sono di particolare rilevanza ai fini del controllo di gestione^[33]:

- **contabilità generale:** consiste di dati consuntivi riguardo movimentazioni monetarie e finanziarie per il rilevamento del risultato economico dell'organizzazione, i documenti prodotti sono tipicamente rivolti verso l'esterno, infatti uno tra i più importanti output di questa funzione è il bilancio d'esercizio;
- **contabilità analitica:** o industriale (v. paragrafo successivo), è lo strumento per la rilevazione di dati riguardanti la gestione aziendale interna, può ad esempio consistere in una rielaborazione dei dati di contabilità generale finalizzata alla rilevazione di particolari situazioni interne a supporto di processi decisionali;
- **budget:** contiene dati a carattere economico – finanziario riguardo gli obiettivi aziendali di ogni area nel relativo periodo, può quindi essere considerato parte integrante della programmazione;

- **reporting:** elaborazione di documenti sulla base degli output prodotti dalle funzioni illustrate nei punti precedenti per evidenziare le variabili chiave;
- **analisi scostamenti:** analisi ed interpretazione delle variazioni tra i dati programmati e quelli consuntivi;

Classificazione dei costi

Nell'ambito della contabilità industriale è di particolare importanza analizzare i costi e classificarli in modo da poter successivamente orientare i prezzi e definire le scelte strategiche ed operative da realizzare. A seconda degli elementi che compongono i costi si possono delineare i seguenti criteri di classificazione e tipologie di costi^[34]:

- **variabilità:** si distingue tra
 - *costi fissi o di struttura:* non variano al variare del volume di produzione e di vendita, ad esempio affitti, manutenzione e assicurazioni;
 - *costi variabili:* variano in base alle quantità prodotte e vendute, ad esempio fanno parte di questa categoria i costi sostenuti per la manodopera, per le materie prime e per le manutenzioni;
- **attribuzione:** si distingue tra
 - *costi diretti:* possono essere imputati al prodotto finale, come ad esempio le materie prime e la manodopera diretta;
 - *costi indiretti:* non sono direttamente collegati al prodotto ma a fattori di produzione, ad esempio appartengono a questa categoria le manutenzioni dei mezzi di produzione e gli affitti;

- **momento di determinazione:**
 - *preventivi:* il costo è stato previsto in fase di programmazione;
 - *consuntivi:* si viene a conoscenza dell'ammontare del costo a lavoro svolto;

- **controllo direzionale:**
 - *controllabili:* influenzabili dalle decisioni dei responsabili ai vari livelli nell'organizzazione entro un certo periodo di tempo;
 - *non controllabili:* non influenzabili dalle azioni e decisioni dei responsabili d'area;

- **tipicità:**
 - *standard:* costo teorico che si sostiene in condizioni di funzionamento dell'impresa;

- **effettiva manifestazione numeraria:**
 - *effettivi:* hanno una concreta manifestazione numeraria, possono quindi essere rilevati direttamente dalla contabilità generale;
 - *figurativi:* gli altri costi non rilevabili dalla contabilità generale come ad esempio la remunerazione dei titolari d'impresa;

3.4 Gli studi professionali

Le tecniche di controllo di gestione per gli *studi professionali* si basano sulle caratteristiche peculiari delle imprese di servizi. Tra i fattori che tipicamente influenzano il giudizio sull'andamento aziendale vi sono la soddisfazione del cliente, la qualità del servizio percepita da quest'ultimo e la tempistica (ed esempio puntualità, tempestività delle consegne e tempi di realizzazione), non si può comunque non considerare i fattori economico – finanziari tipici dell'impresa industriale^[35].

Generalmente i servizi forniti dagli *studi professionali* non sono dei servizi di massa ma sono pensati *ad hoc* per il cliente. Infatti gran parte del tempo delle risorse umane viene impiegato nel rapporto col cliente e tali risorse godono di una certa autonomia data l'eterogeneità dei possibili servizi da fornire. Alla luce di queste caratteristiche e tenendo conto dell'interfunzionalità delle risorse si capisce che l'organizzazione migliore prevede l'identificazione di aree strategiche di servizio che si comportano come vere e proprie aziende autonome che lavorano per progetti^[35]:

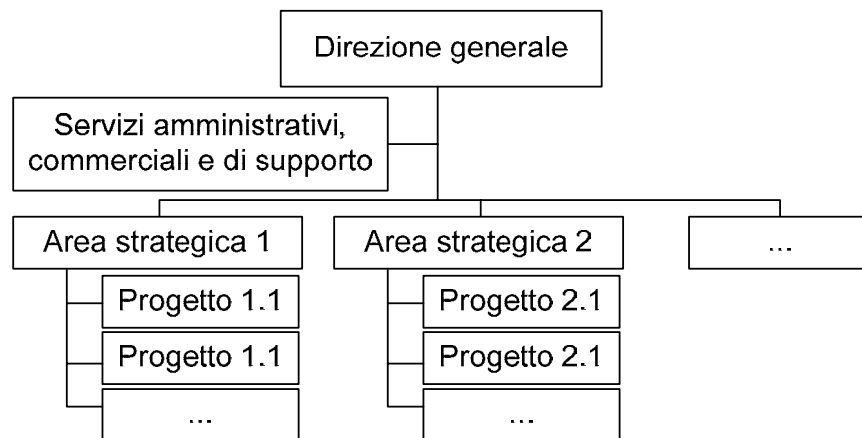


Figura 9: Aree strategiche

La *Figura 10* illustra le fasi generalmente eseguite da un professionista per la gestione di ogni progetto^[36]:

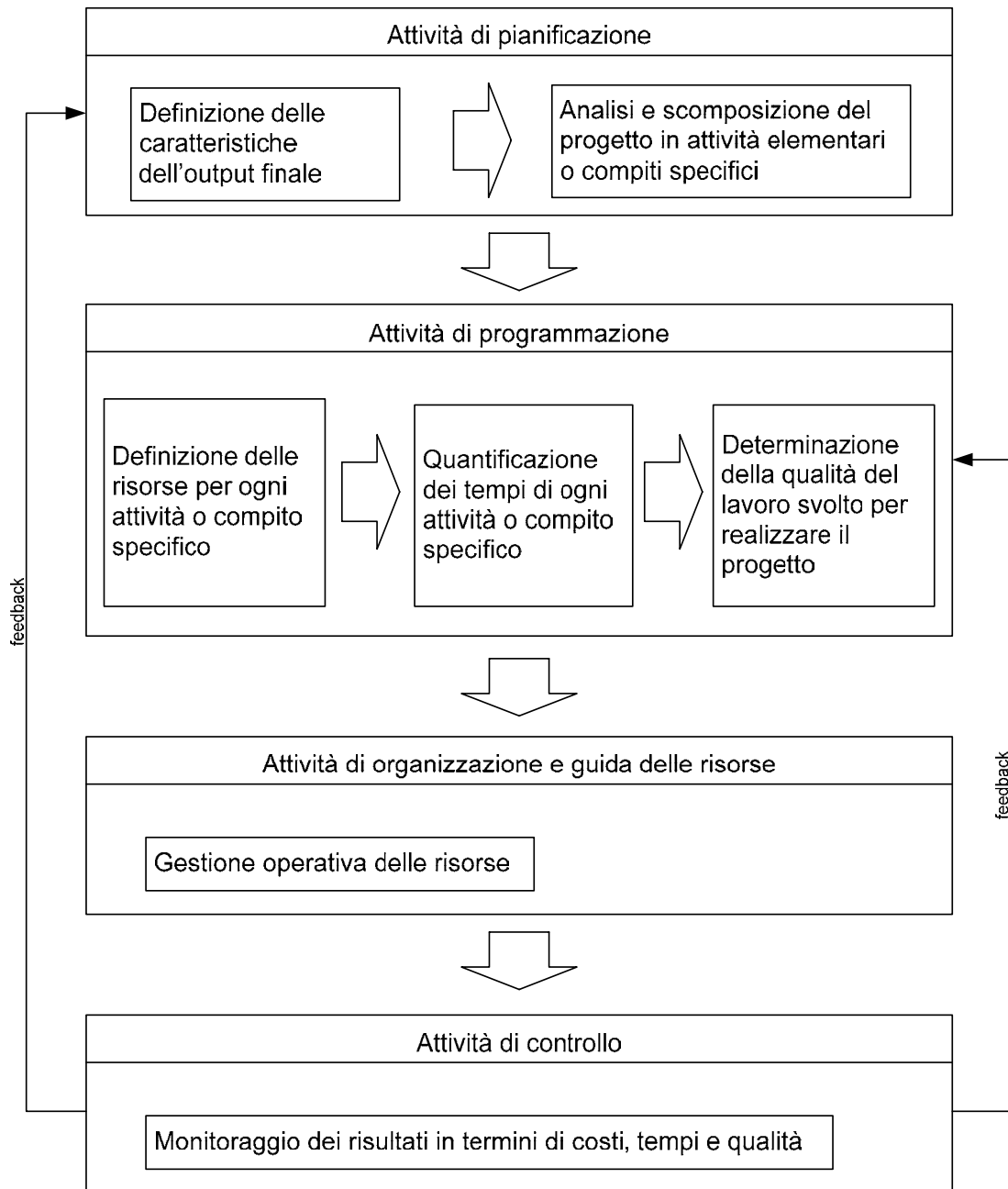


Figura 10: Fasi di gestione di un progetto

Si può notare come tale visione sia in accordo col *sistema di gestione qualità* illustrato in *2.5.2 Progettazione e sviluppo*.

3.5 Requisiti Syndev

Syndev deve fornire delle funzionalità per permettere alla direzione di effettuare il *controllo di gestione*. In particolare è stato richiesto di generare una serie di report riguardo le entità descritte nel capitolo precedente (v. 2.6 *Requisiti Syndev*) e per questo è necessario riferirsi alle tipologie descritte in 3.3 *Gli strumenti del controllo di gestione* e aggiungere alle caratteristiche delle commesse il seguenti attributi:

- **attività:** momento in cui si valutano i costi, è stato scelto di inserire tale attributo tra le tabelle (v. 2.6.11.1 *Tabelle*):
 - a preventivo;
 - a consuntivo;
- **tipologia:** l'amministrazione attribuisce alla commessa la categoria adatta in modo da analizzarne correttamente l'andamento, tale attributo ammette i valori:
 - diretta di produzione:
 - commessa standard di produzione come descritto in 1.f.ii.1 *Processo di sviluppo*;
 - indiretta:
 - risorse non allocate o produttive, manutenzioni e supporto commerciale;
 - indiretta di produzione:
 - attività indirette ma attinenti alla produzione;
 - assenza:
 - ferie, malattie o maternità.

Inoltre tra le *tabelle* si deve registrare il *costo orario* di ogni risorsa, può essere *interno* o *esterno* ed è relativo ad un intervallo temporale mensile (v. 2.6.11.1 *Tabelle*).

I report da fornire prevedono l'elaborazione e aggregazione dei dati di fatturazione, prevista o effettiva, e dei giorni, pianificati o impiegati, riguardanti le commesse, le risorse e i clienti. Sono richiesti i seguenti riepiloghi (v. *Appendice A*):

- dettagli commessa accessibile da lista commesse;
- riepilogo commesse interne;

- riepilogo commesse esterne;
- riepilogo commesse indirette;
- riepilogo totale;
- riepilogo risorse;
- dettaglio risorsa.

Per avere una rappresentazione visuale degli scostamenti si introduce un **semaforo** colorato a seconda dello scostamento dei risultati rispetto alla programmazione:

Colore	Valore percentuale
Verde	$X \geq 20\%$
Giallo-verde	$20\% > X \geq 10\%$
Giallo	$10\% > X > 0\%$
Arancione	$-10\% \leq X < 0\%$
Rosso	$-10\% < X$

Figura 11: Semaforo

3.5.1 Lista commesse

Attributi di ogni riga della lista:

- *Codice*
- *Titolo*
- *Competenza*
- *Stato commessa*
- *Stato sviluppo*
- *Area*
- *Cliente*
- *Attività*
- *Ambiente*
- *Responsabile progetto*
- *Procedura*
- *Giorni offerta*
- *Importo offerta*
- *% realizzazione*
- *Giorni realizzazione = giorni offerta * %realizzazione*

- *Importo realizzazione = importo offerta * %realizzazione*
- *Giorni effettuati =*

$$\sum_{\text{cartellino}} \text{ore lavorate} / 8$$
- *Costi sostenuti =*

$$\sum_{\text{cartellino}} \text{ore lavorate} * \text{costo orario risorsa}$$
- *Delta giorni = giorni realizzazione – giorni effettuati*
- *Delta ricavo = importo realizzazione – costi sostenuti*
- *%giorni = delta giorni/giorni realizzazione*
- *%importo = delta ricavo/importo realizzazione*
- *Semaforo Delta ricavo*

Righe della lista:

- *Commesse*
- *Totale commesse visualizzate*

Filtri di ricerca:

- *Anno*
- *Dal*
- *Al*
- *Competenza*
- *Stato commessa*
- *Stato sviluppo*
- *Area*
- *Cliente*
- *Attività*
- *Ambiente*
- *Responsabile progetto*
- *Procedura*

3.5.2 Dettagli commessa

La visualizzazione dei dettagli commessa nell'ambito del controllo prevede l'esposizione di:

- *dati offerta (v. 2.6.3 Offerta);*
- *totali fatturazione prevista ed effettiva (v. 2.6.4 Fatturazione);*
- *percentuale realizzazione calcolata;*
- *percentuale realizzazione stimata modificabile;*
- *data stima modificabile;*
- *riepilogo attività svolta, per ogni risorsa:*
 - *ore lavorate;*
 - *giorni lavorati = ore lavorate/8;*
 - *Costi =*
$$\sum_{\text{cartellino}} (\text{ore lavorate} * \text{costo orario risorsa})$$
 - *%giorni = giorni lavorati/giorni totali commessa*
 - *%fatturato = %giorni * importo realizzato*
- *totali dati del punto precedente;*

3.5.3 Riepilogo commesse interne

Attributi di ogni riga della lista:

- *Cliente*
- *Attività*
- *Ambiente*
- *Responsabile progetto*
- *Giorni offerta =*
$$\sum_{\text{commesse di area interna}} \text{giorni offerta}$$
- *Importo offerta =*
$$\sum_{\text{commesse di area interna}} \text{importo offerta}$$
- *Giorni effettuati attività =*
$$\sum_{\text{commesse di area interna}} \left(\sum_{\text{cartellino}} \text{ore lavorate} / 8 \right)$$
- *Costi sostenuti attività =*
$$\sum_{\text{commesse di area interna}} \left(\sum_{\text{cartellino}} \text{ore lavorate} * \text{costo orario risorsa} \right)$$

- *Giorni fatturato* =
$$\sum_{\text{commesse di area interna}} \left(\sum_{\text{cartellino}} \text{giorni offerta} * \%realizzazione \right)$$
- *Importo fatturato* =
$$\sum_{\text{commesse di area interna}} \left(\sum_{\text{cartellino}} \text{importo offerta} * \%realizzazione \right)$$
- *Delta giorni* = *giorni fatturato* – *giorni effettuati*
- *Delta ricavo* = *importo fatturato* – *costi sostenuti*
- *%giorni* = *delta giorni*/*giorni offerta*
- *%importo* = *delta ricavo*/*importo offerta*
- *Semaforo delta ricavo*
- *Percentuali per livello (v.righe della lista):*
 - *Responsabile progetto/Cliente*
 - *Responsabile progetto*
 - *Ambiente*
 - *Attività*
 - *Cliente*

Righe della lista:

- *Commesse ordinate per responsabile di progetto*
- *Totale commesse per responsabile di progetto*
- *Totale commesse per ambiente*
- *Totale commesse per attività*
- *Totale commesse per cliente*
- *Totale produzione diretta*
- *Totale costi indiretti*
- *Totale costi diretti + indiretti*

Filtri:

- *Anno*
- *Dal*
- *Al*
- *Cliente*
- *Attività*
- *Ambiente*
- *Responsabile progetto*

3.5.4 Riepilogo commesse esterne

Attributi di ogni riga della lista:

- *Cliente*
- *Attività*
- *Ambiente*
- *Procedura*
- *Giorni offerta* =
$$\sum_{\text{commesse di area esterna}} \text{giorni offerta}$$
- *Importo offerta* =
$$\sum_{\text{commesse di area esterna}} \text{importo offerta}$$
- *Giorni effettuati attività* =
$$\sum_{\text{commesse di area esterna}} \left(\sum_{\text{cartellino}} \text{ore lavorate} / 8 \right)$$
- *Costi sostenuti attività* =
$$\sum_{\text{commesse di area esterna}} \left(\sum_{\text{cartellino}} \text{ore lavorate} * \text{costo orario risorsa} \right)$$
- *Giorni fatturato* =
$$\sum_{\text{commesse di area esterna}} \left(\sum_{\text{cartellino}} \text{giorni offerta} * \%realizzazione \right)$$
- *Importo fatturato* =
$$\sum_{\text{commesse di area esterna}} \left(\sum_{\text{cartellino}} \text{importo offerta} * \%realizzazione \right)$$
- *Delta giorni* = *giorni fatturato* – *giorni effettuati*
- *Delta ricavo* = *importo fatturato* – *costi sostenuti*
- *%giorni* = *delta giorni*/*giorni offerta*
- *%importo* = *delta ricavo*/*importo offerta*
- *Semaforo delta ricavo*
- *Percentuali per livello (v.righe della lista):*
 - Procedura
 - Cliente
 - Ambiente
 - Attività

Righe della lista:

- *Commesse ordinate per cliente*
- *Totale commesse per cliente*
- *Totale commesse per ambiente*
- *Totale commesse per attività*
- *Totale produzione diretta*
- *Totale costi indiretti*
- *Totale costi diretti + indiretti*

Filtri di ricerca:

- *Anno*
- *Dal*
- *Al*
- *Cliente*
- *Attività*
- *Ambiente*
- *Responsabile progetto*

3.5.5 Riepilogo commesse indirette

Attributi di ogni riga della lista:

- *Codice*
- *Descrizione*
- *Risorsa*
- *Giorni*
- *Costo*
- *Percentuali:*
 - *Risorsa*
 - *Commessa*
 - *Area*
 - *Dirette/indirette*

Righe della lista:

- *Risorse ordinate e suddivise per commessa*
- *Totale commessa*
- *Totale area*
- *Totale indirette di produzione*
- *Totale indirette*
- *Totale*

Filtri di ricerca:

- *Anno*
- *Dal*
- *Al*
- *Area*

3.5.6 Riepilogo totale

Attributi di ogni riga della lista:

- *Area*
- *Cliente*
- *Attività*
- *Ambiente*
- *Giorni offerta*
- *Importo offerta*
- *Giorni effettuati*
- *Costi sostenuti*
- *Giorni fatturati*
- *Importo fatturato*
- *Delta giorni = giorni fatturati – giorni effettuati*
- *Delta ricavo = Importo fatturato – costi sostenuti*
- *Delta giorni/giorni fatturati*
- *Delta ricavo/importo fatturato*
- *Semaforo delta ricavo*
- *Percentuali per livello (v.righe della lista):*

Righe della lista:

- *Commesse interne*
 - *Commesse per cliente/ambiente*
 - *Totale cliente*

- o *Totale ambiente*
- o *Totale attività*
- o *Totale interne*
- o *Totale varie interne*
- o *Totale area interne*
- *Commesse esterne*
 - o *Commesse per cliente/ambiente*
 - o *Totale cliente*
 - o *Totale ambiente*
 - o *Totale attività*
 - o *Totale esterne*
 - o *Totale varie esterne*
 - o *Totale area interne*
 - o *Totale varie indirette*
 - o *Totale*
- *Commesse per cliente*
- *Commesse per attività*
- *Commesse per ambiente*

Filtri di ricerca:

- *Anno*
- *Dal*
- *Al*

3.5.7 Riepilogo risorse

Attributi di ogni riga della lista:

- *Nome risorsa*
- *Ore dirette*
- *Costi diretti*
- *Ore indirette*
- *Costi indiretti*
- *Totale giorni lavorati*
- *Totale costo*
- *Assenze*

- *Giorni fatturati* =

$$\sum_{commesse} \text{giorni offerta} * \%realizzazione$$

Con:

$$\%realizzazione = \begin{cases} \left(\frac{\text{ore lavorate}}{\text{ore totali}} \right) * \%realizzazione commessa & \text{se commessa aperta} \\ \left(\frac{\text{ore lavorate}}{\text{ore totali}} \right) & \text{se commessa chiusa} \end{cases}$$

- *Importo fatturato* =

$$\sum_{commesse} \text{importo offerta} * \%realizzazione$$

- *Delta giorni* = *giorni offerta* – *giorni fatturati*
- *Delta fatturato* = *importo fatturato* – *totale costo*
- *Delta giorni/giorni fatturati*
 - *semaforo*
- *delta fatturato/importo fatturato*
 - *semaforo*
- *Percentuali sul totale:*
 - *Giorni svolti*
 - *Costo*
 - *Giorni fatturati*
 - *Importo fatturato*

Righe della lista:

- *Risorsa*
- *Totale*

Filtri di ricerca:

- *Anno*
- *Dal*
- *Al*
- *Area cliente*
- *Attività*
- *Ambiente*
- *Responsabile di progetto*
- *Procedura*

3.5.8 Dettaglio risorsa

Riepilogo:

- *Ore produzione*
- $\% \text{produzione} = \text{ore produzione} / \text{totale ore}$
- *Ore commesse varie*
- *Ore assenze*
- *Totale ore lavorative*
- $\% \text{ore lavorative} = (\text{ore produzione} + \text{ore commesse varie}) / \text{totale ore}$
- *Totale ore*
- *Percentuali ore assenze, ore commesse varie dirette, ore commesse varie indirette*

Dettaglio, per ogni commessa e totale:

- *Presenze:*
 - *Ore svolte*
 - *Costo*
 - *Ore fatturate*
 - *Importo fatturato*
 - $\text{Delta ore} = \text{ore fatturate} - \text{ore svolte}$
 - $\text{Delta fatturato} = \text{importo fatturato} - \text{costo}$
 - $\% \text{ore} = \text{delta ore} / \text{ore fatturate}$
 - $\% \text{fatturato} = \text{delta fatturato} / \text{importo fatturato}$
 - *%sul totale*
- *Assenze:*
 - *Ore*
 - *%sul totale*

3.5.9 Altri report ed export

Altri report che *Synde* deve fornire sono i riepiloghi dei cartellini mensili di ogni risorsa e delle presenze.

Per *export* si intende la stampa di report su file in modo che possano essere elaborati esternamente al sistema, ad esempio con fogli di calcolo.

Inoltre *Synde* fornirà la possibilità di stampare il piano bianco e consolidato di ogni commessa.

3.5.10 Casistica commesse

Per definire quali commesse considerare nell'utilizzo delle funzioni di controllo di gestione e se considerarne solo i costi, i ricavi o entrambi è stato necessario elaborare una casistica delle commesse in base all'anno di emissione dell'offerta, di svolgimento delle attività, di fatturazione e di competenza.

Nella pagina seguente sono elencati i vari casi delle commesse di pertinenza dell'anno in corso.

OFFERTA	ATTIVITA'	FATTURAZIONE	COMPETENZA DI BILANCIO	bilancio	Visualizzazione GESTIONE	Visualizzazione AMMINISTRAZIONE CONTROLLO	Visualizzazione
							NB : no anno - tutti i dati
1	anno in corso	anno in corso	anno in corso	bilancio anno in corso	anno in corso	anno in corso	anno in corso
2.01	anno precedente	anno in corso	anno precedente	bilancio anno precedente costi e ricavi bilancio anno in corso - non imputata	anno precedente	anno in corso	anno precedente
2.02	anno precedente	anno in corso	anno in corso	bilancio anno precedente - solo costi bilancio anno in corso - solo ricavi	anno precedente	anno in corso	anno precedente - solo i costi anno precedente anno anno in corso - solo i ricavi
3.01	anno precedente / in corso	anno in corso	anno precedente	bilancio anno precedente - costi attività anno precedente e ricavi bilancio anno in corso - solo costi attività anno in corso	anno in corso	anno in corso	anno precedente - costi anno precedente e ricavi anno precedente anno anno in corso - costi anno in corso
3.02	anno precedente / in corso	anno in corso	anno in corso	bilancio anno precedente - costi attività anno precedente bilancio anno in corso - costi attività anno in corso e ricavi	anno in corso	anno in corso	anno precedente - costi anno precedente anno anno in corso - costi anno in corso e ricavi anno in corso
4.01	anno in corso	anno in corso	anno precedente	bilancio anno precedente - solo ricavi bilancio anno in corso - solo costi attività anno in corso	anno in corso	anno in corso	anno precedente - ricavi anno precedente anno anno in corso - solo costi anno in corso
4.02	anno in corso	anno in corso	anno in corso	bilancio anno precedente - non imputata bilancio anno in corso - costi attività anno in corso e ricavi	anno in corso	anno in corso	anno anno in corso
5.01	anno in corso	anno prossimo	anno in corso	bilancio anno in corso - costi e ricavi bilancio anno prossimo - non imputata	anno in corso	anno in corso	anno anno in corso
5.02	anno in corso	anno prossimo	anno prossimo	bilancio anno in corso - solo costi bilancio anno prossimo - solo ricavi	anno in corso	anno in corso	anno in corso - solo costi anno in corso anno 2008 - solo i ricavi
6.01	anno in corso	anno prossimo	anno prossimo	bilancio anno in corso - non imputata bilancio anno prossimo - costi e ricavi	anno prossimo	anno prossimo	anno prossimo
6.02	anno in corso	anno prossimo	anno in corso	bilancio anno in corso - ricavi bilancio anno prossimo - solo i costi	anno prossimo	anno prossimo	anno anno in corso - solo i ricavi anno prossimo - solo i costi anno prossimo
7	anno in corso	anno prossimo	anno in corso	bilancio anno in corso - costi attività anno in corso e ricavi bilancio anno prossimo - non imputata	anno prossimo	anno prossimo	anno in corso - costi e ricavi

Figura 12: Esame commesse di pertinenza dell'anno in corso

4.1.2 Accesso

All'avvio *Synde* mostra una schermata d'accesso per l'inserimento delle informazioni di autenticazione (username e password) e riconosce il tipo di utente. A meno di un errore una risorsa semplice accede alle funzioni di cartellino mentre un responsabile o amministratore al menù principale che presenta gli ambiti di operazioni effettuabili. Dal menù principale si può accedere alle funzioni di produzione pianificazione, amministrazione, controllo, gestione tabelle, export e compilazione dei cartellini delle altre risorse.

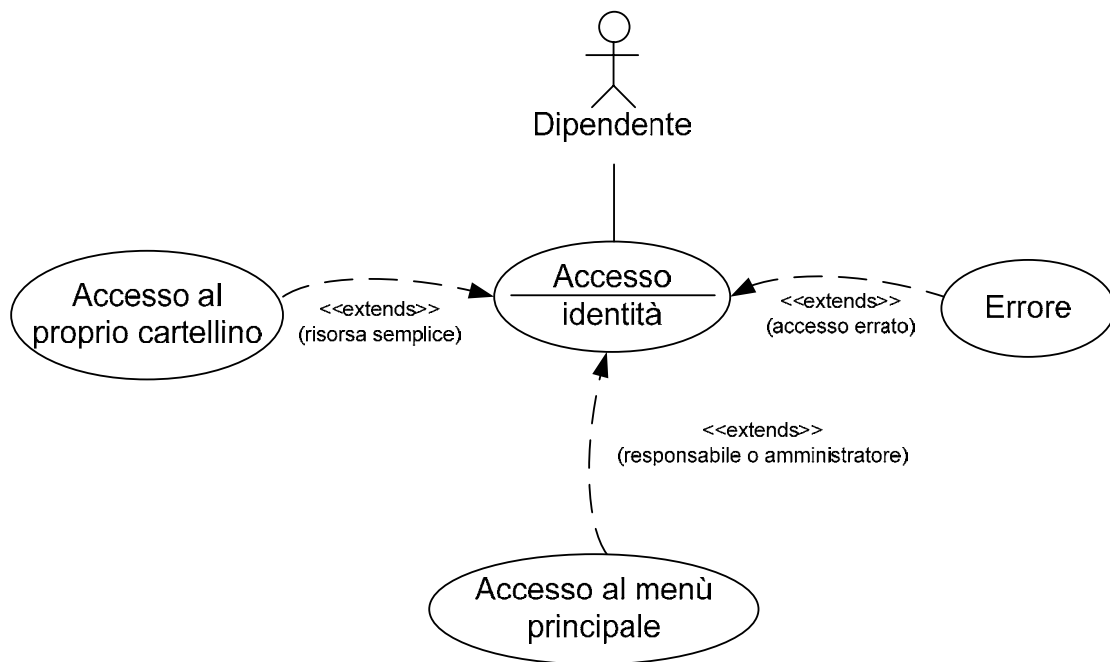


Figura 14: Use case – accesso

4.1.3 Cartellino

Questa sezione prevede la visualizzazione di un calendario dal quale è possibile accedere al giorno nel quale si vuole compilare il proprio cartellino. L'amministratore può accedere a tali funzioni anche per conto di altre risorse tramite una lista. Un altro servizio fornito è la generazione di report presenze.

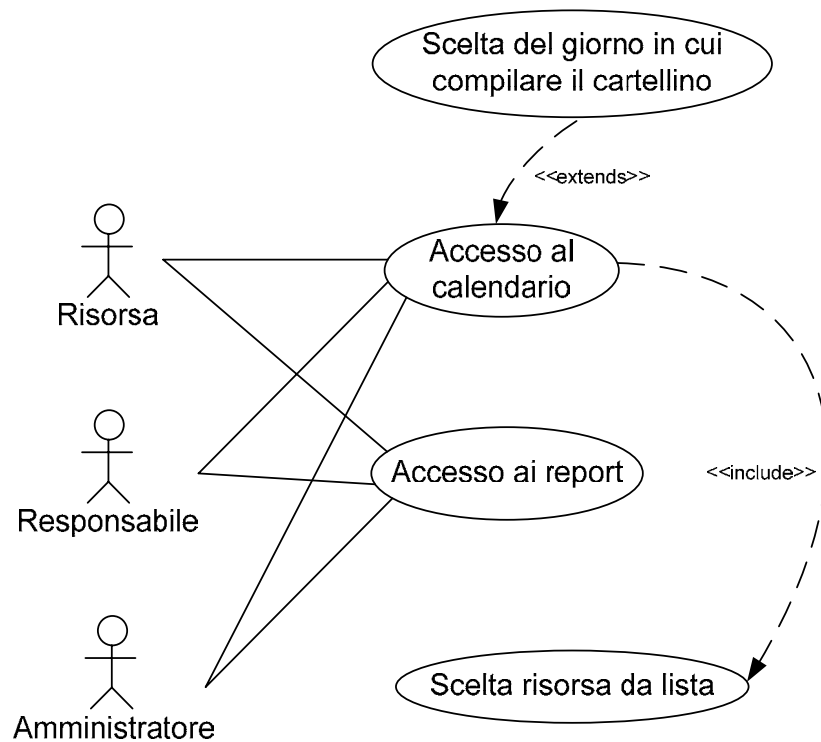


Figura 15: Use case – cartellino

4.1.4 Produzione e pianificazione

La parte di pianificazione prevede la presentazione dei diagrammi di Gantt (v. 2.6.9 *Gantt chart*), dalla produzione si accede al form per l'inserimento di una nuova commessa o alla lista delle commesse di propria competenza e alle schede di tali liste.

L'inserimento di una nuova commessa prevede l'assegnazione di risorse alle fasi di lavoro, per ottimizzare tale scelta si può accedere direttamente al Gantt chart delle risorse.

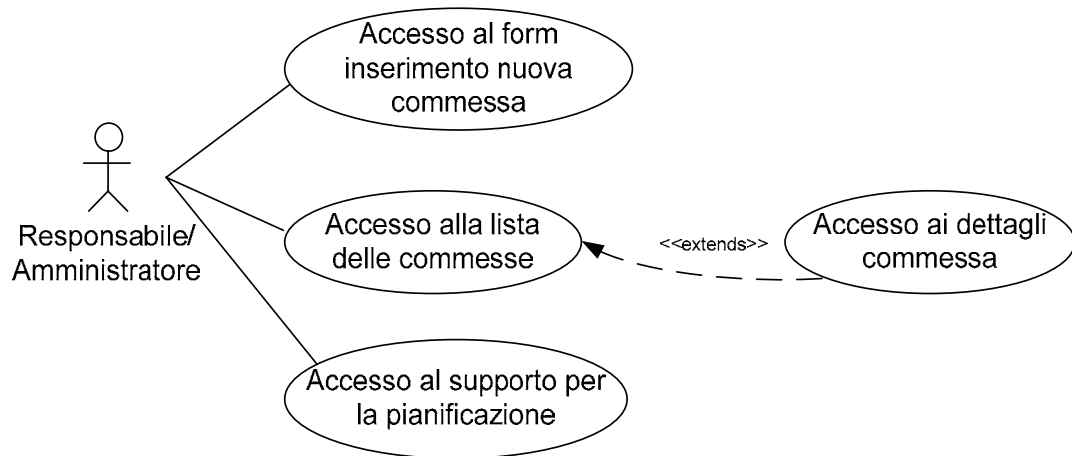


Figura 16: Use case – produzione e pianificazione

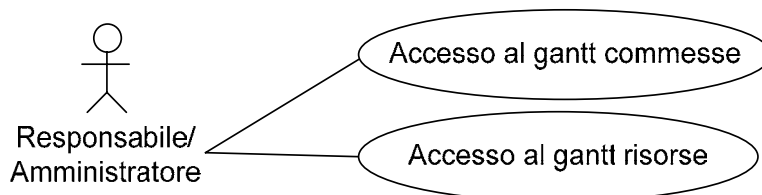


Figura 17: Use case – supporto alla pianificazione

4.1.5 Amministrazione

In fase di amministrazione si può creare una commessa di integrazione o accedere alla lista delle commesse, selezionandone una si possono modificare i dati dell'offerta (v. 2.6.3 *Offerta*), visualizzare il consolidato, gestire le fatturazioni e gli stati dell'offerta.

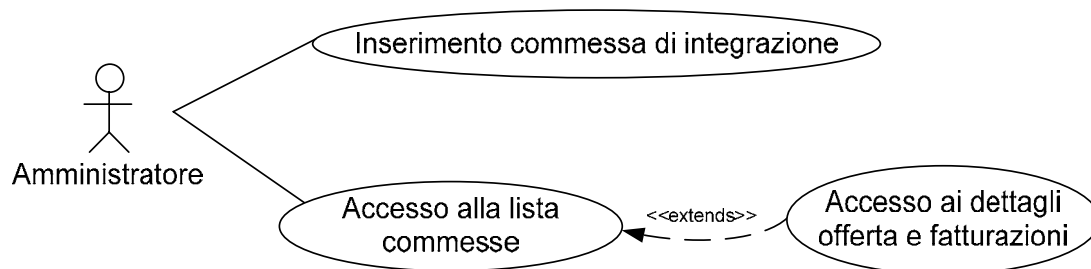


Figura 18: Use case – amministrazione

4.1.6 Controllo

Accesso ai report per il controllo di gestione (v. 3.5 *Requisiti Syndev*).

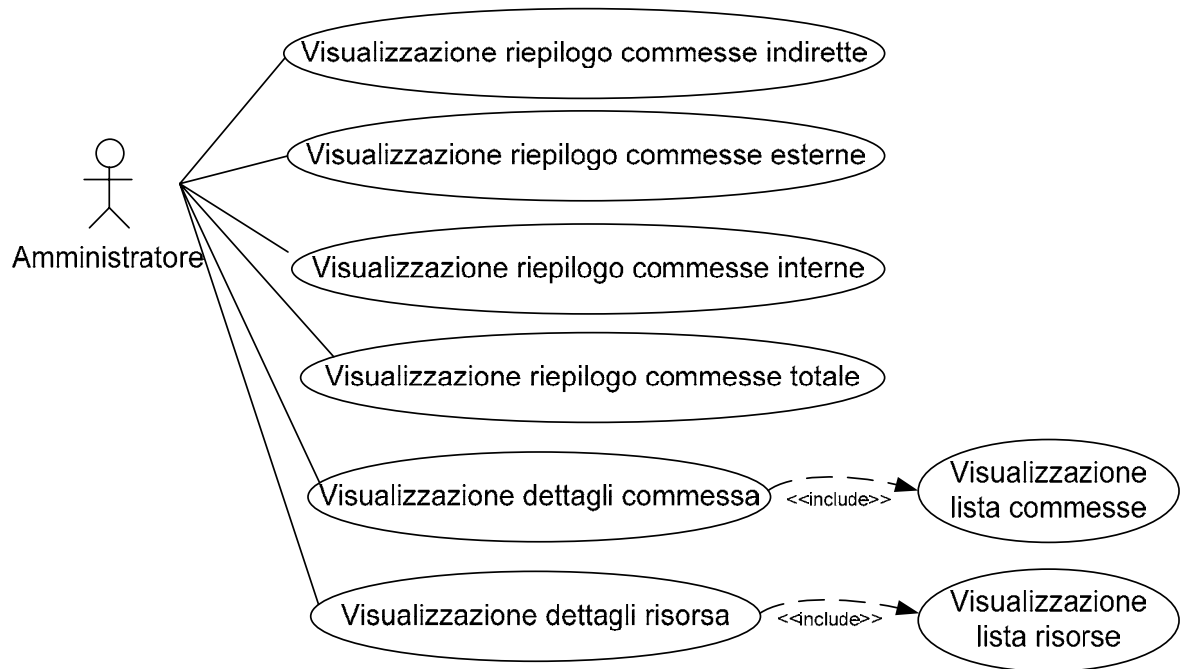


Figura 19: Use case – Controllo

4.1.7 Export

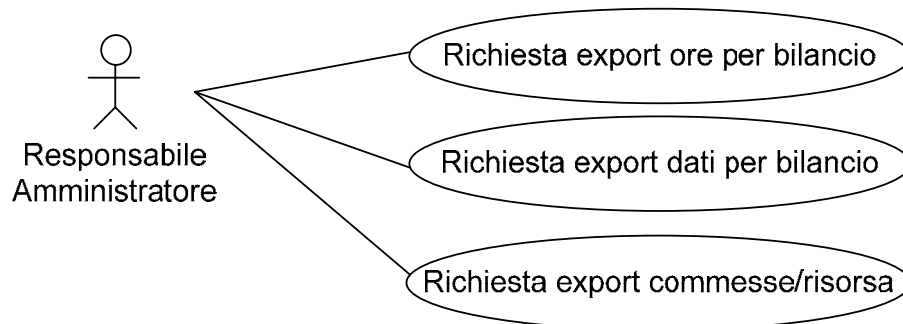


Figura 20: Use case – Export

4.2 Activity diagram

Utilizzando lo strumento degli *activity diagrams* si possono descrivere più nel dettaglio i processi eseguibili tramite le funzioni del software e approfondire i casi d'uso presentati precedentemente. Per fare ciò si ricorrerà all'uso delle "corsie di marcia" per distinguere le attività svolte dall'utente, dal sistema e l'output.

4.2.1 Accesso al sistema

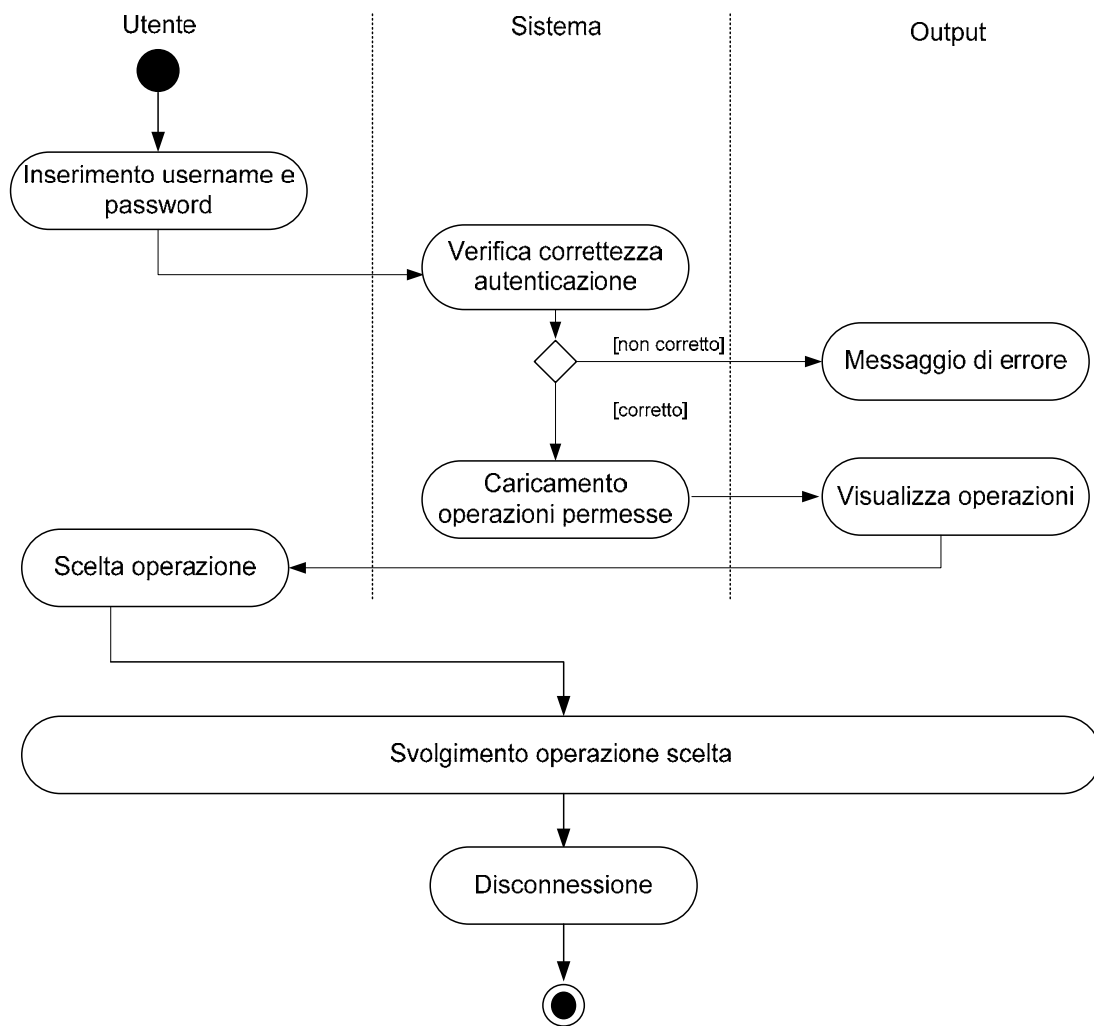


Figura 21: Activity diagrams – accesso

4.2.2 Cartellino

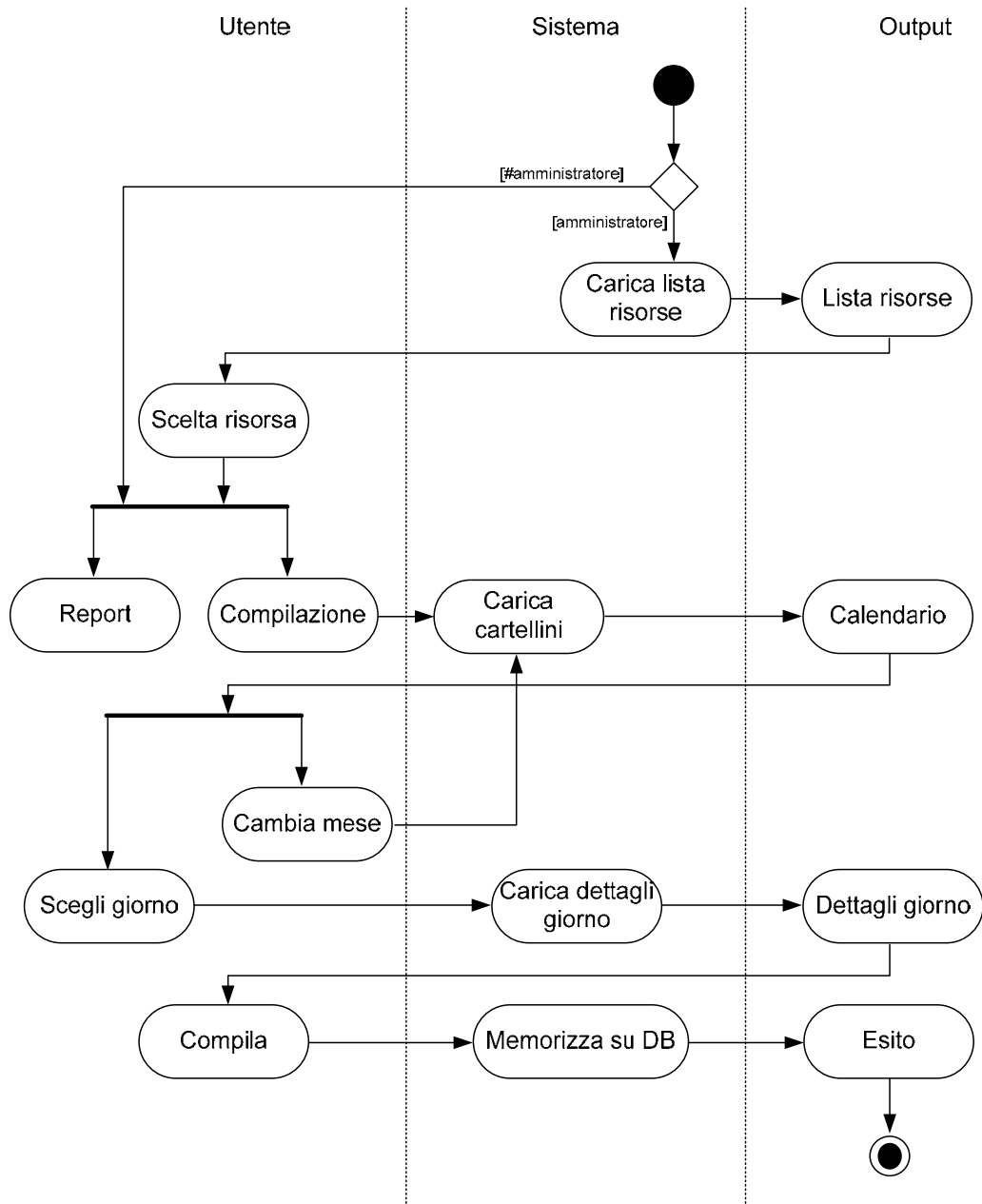


Figura 22: Activity diagrams – cartellino

4.2.3 Produzione e pianificazione

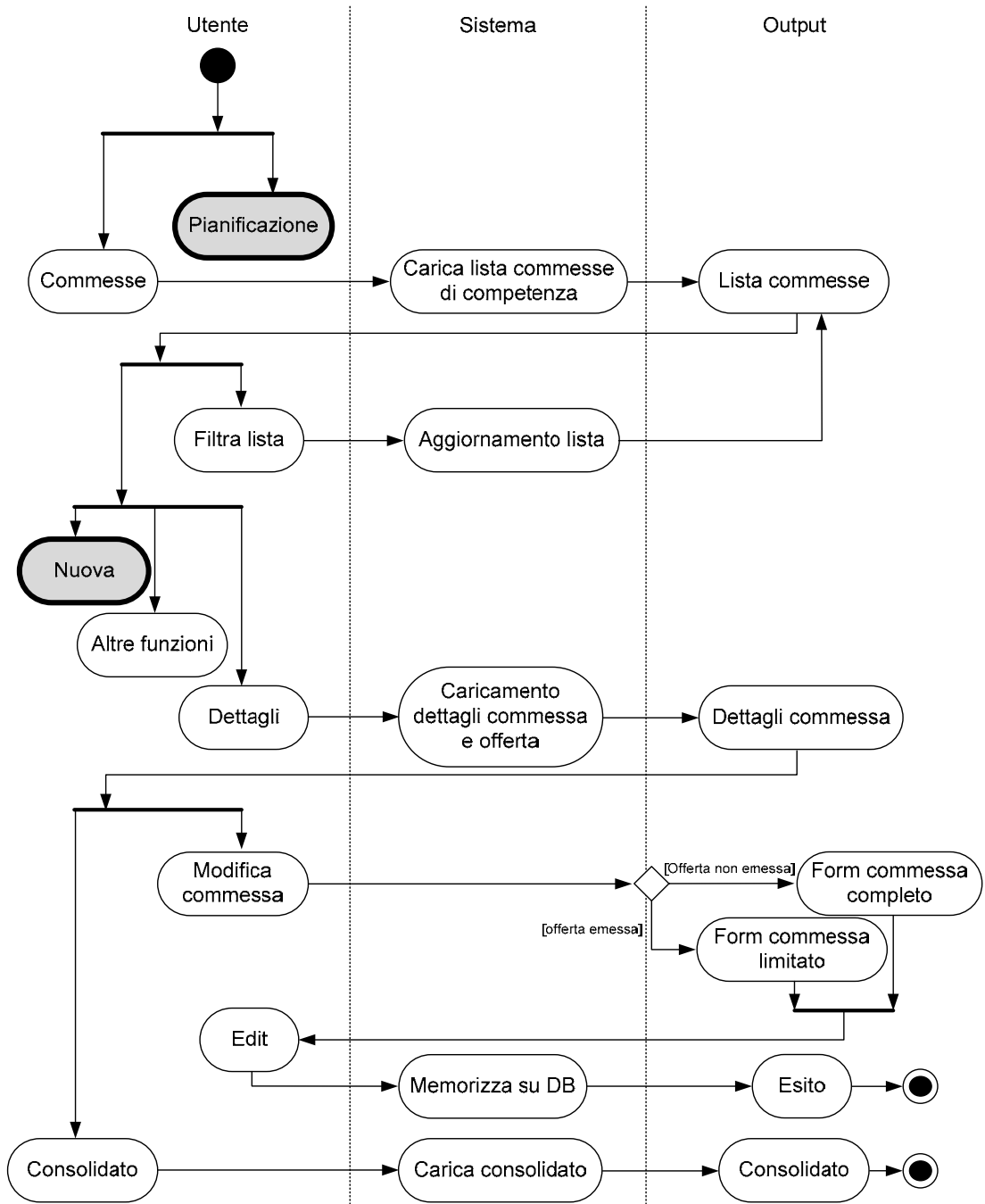


Figura 23: Activity diagrams – produzione e pianificazione

4.2.4 Nuova commessa

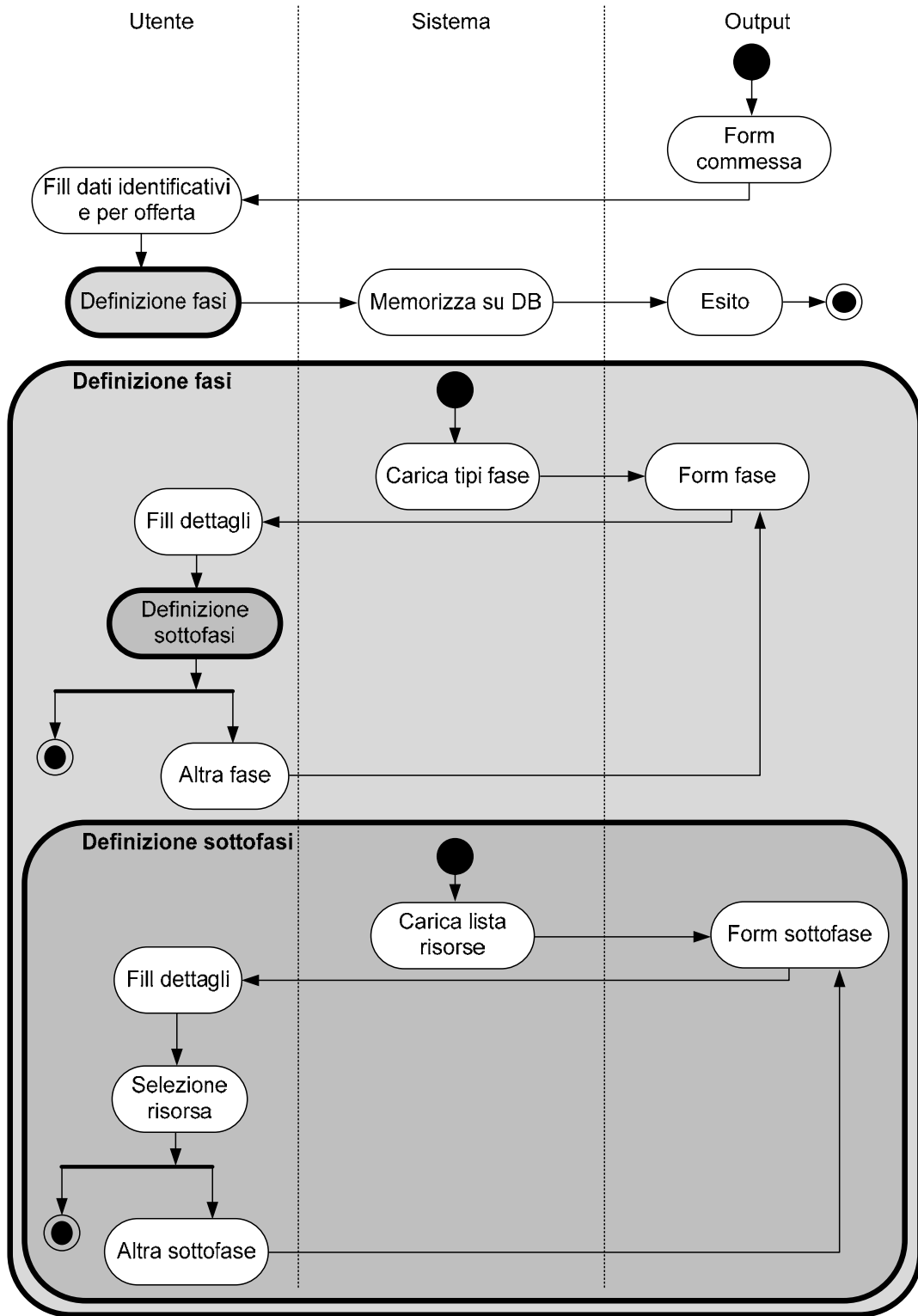


Figura 24: Activity diagram – nuova commessa

4.2.5 Pianificazione

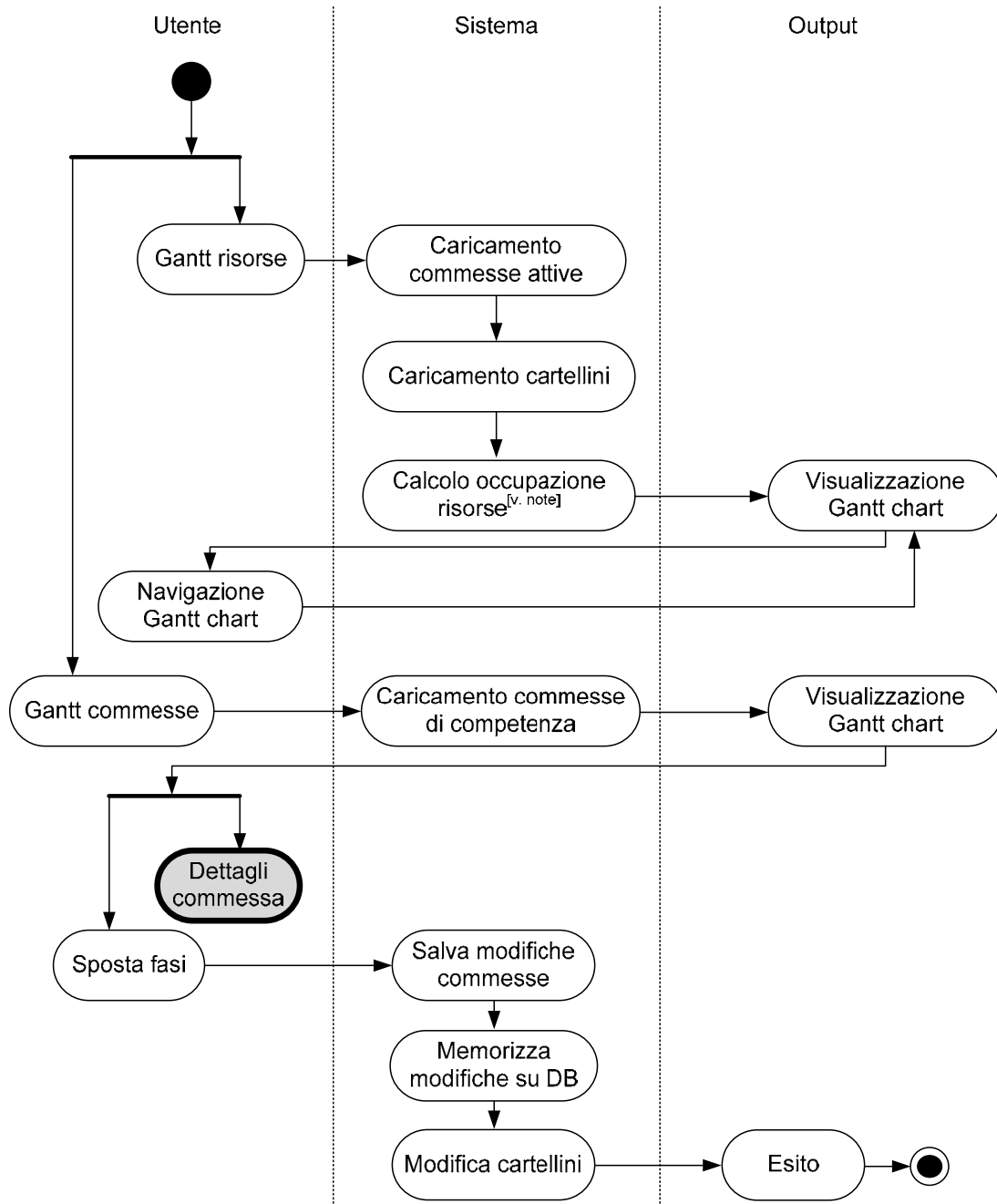


Figura 25: Activity diagram – pianificazione

NOTE:

- L'occupazione della risorsa in un giorno è data da (v. 2.6.9 Gantt chart):

$$\sum_{fasi} \left(\frac{\text{giorni rimasti da lavorare}}{\text{giorni totali}} \right) * 8$$

4.2.6 Amministrazione

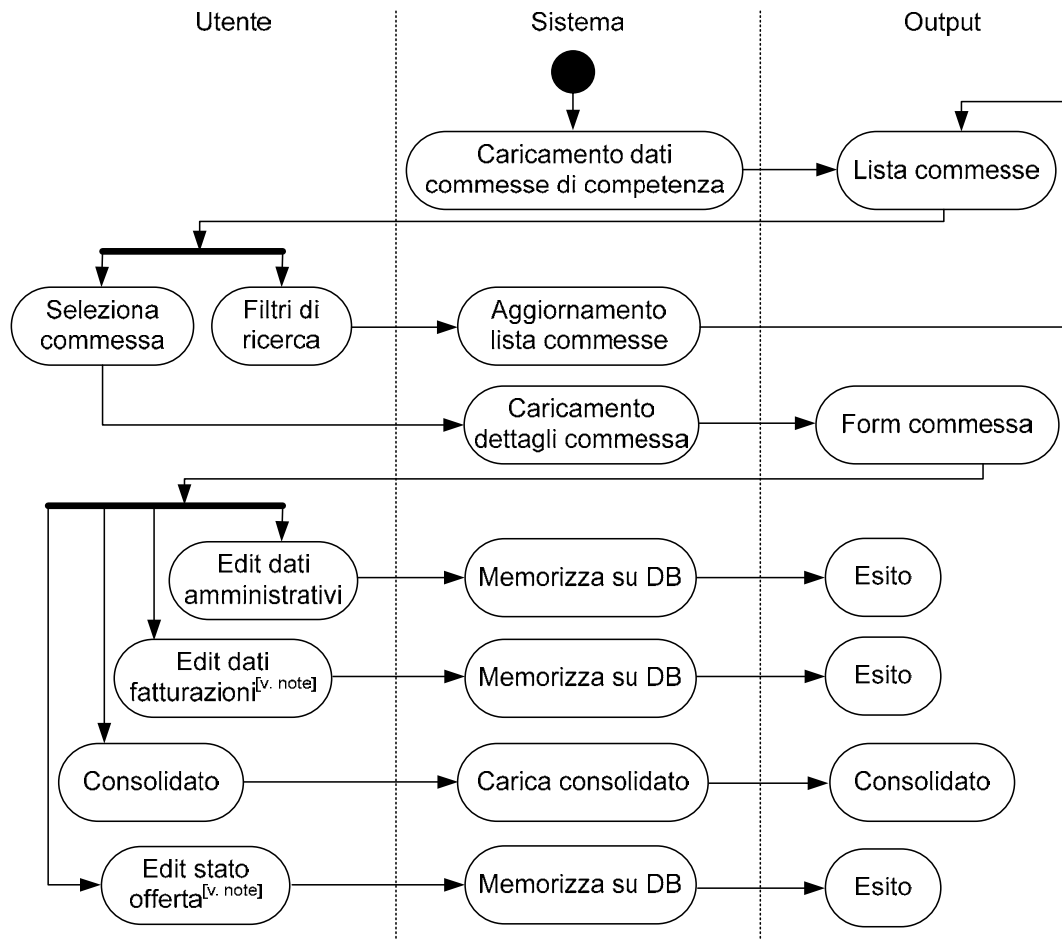


Figura 26: Activity diagram – amministrazione

NOTE:

- Il cambiamento dello stato dell'offerta e la gestione delle fatturazioni segue le regole viste in 2.6.6 *Stati commessa-offerta e interazioni* e il sistema gestisce automaticamente lo stato della commessa corrispondente.

4.2.7 Controllo - lista commesse

Le funzioni di controllo prevedono sostanzialmente che il sistema carichi i dati da elaborare e presentare e l'utente inserisca dei filtri di ricerca come illustrato in 3.5 *Requisiti Syndev*.

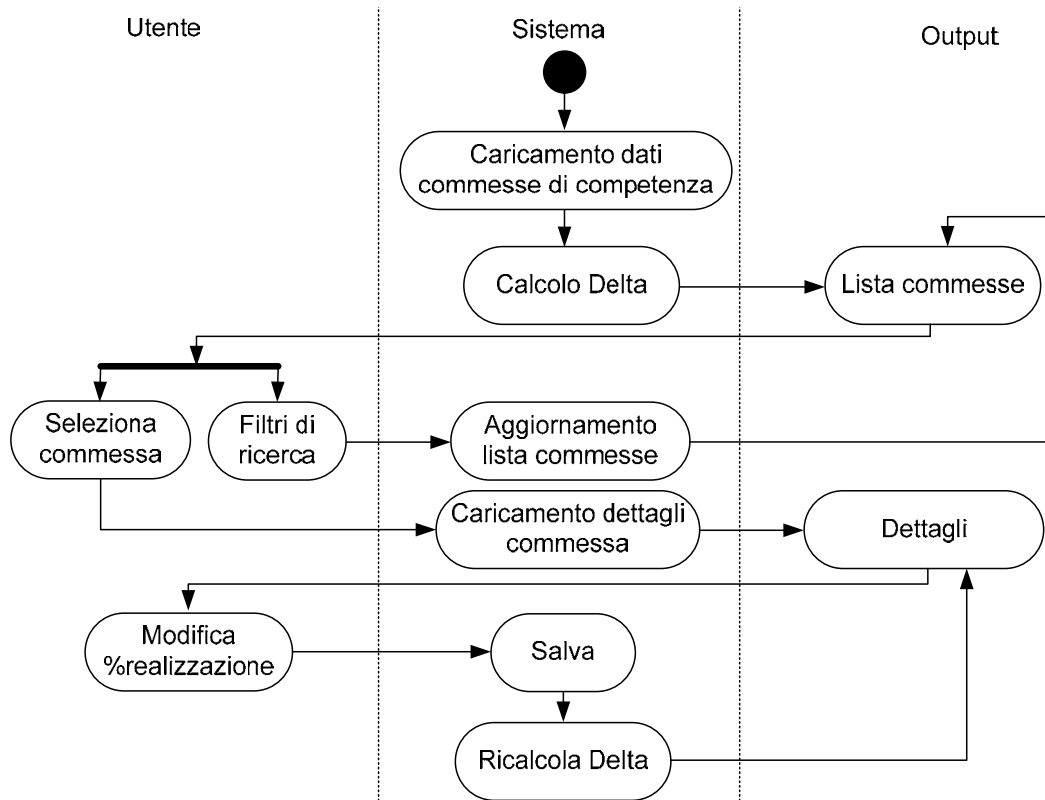


Figura 27: Activity diagram – lista commesse

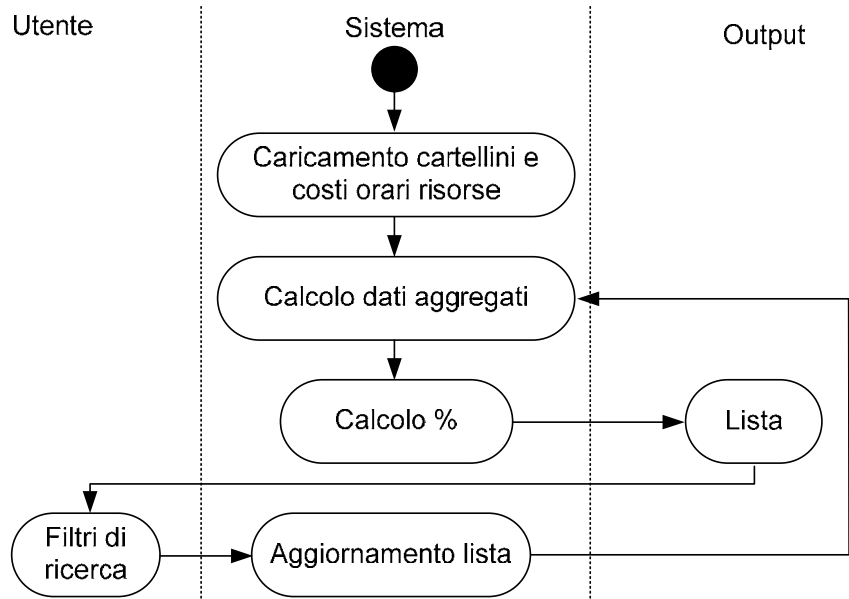


Figura 28: Activity diagrams – riepiloghi

4.3 Class diagram

Con lo strumento dei *class diagram* si possono documentare tutte le classi che compongono il sistema, evidenziarne le caratteristiche (gli attributi e le operazioni) e descrivere le associazioni che le legano.

4.3.1 Packages

Possiamo suddividere le classi che il sistema andrà a gestire in tre Packages:

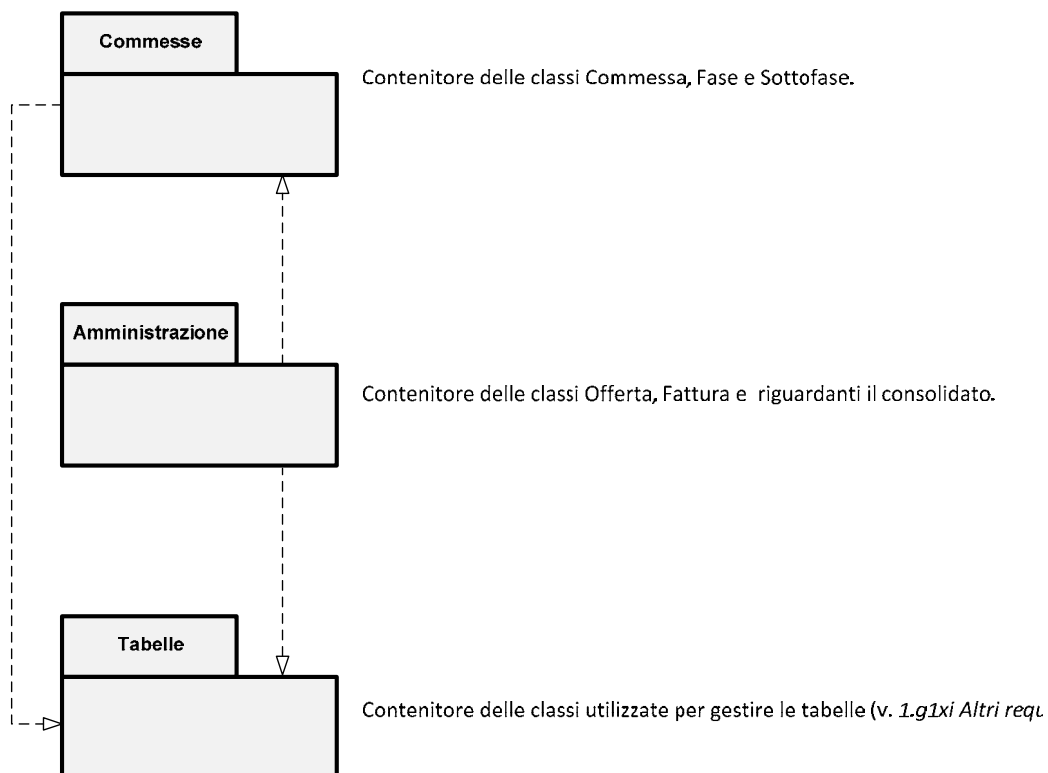


Figura 29: Class diagram – packages

4.3.2 Package commesse

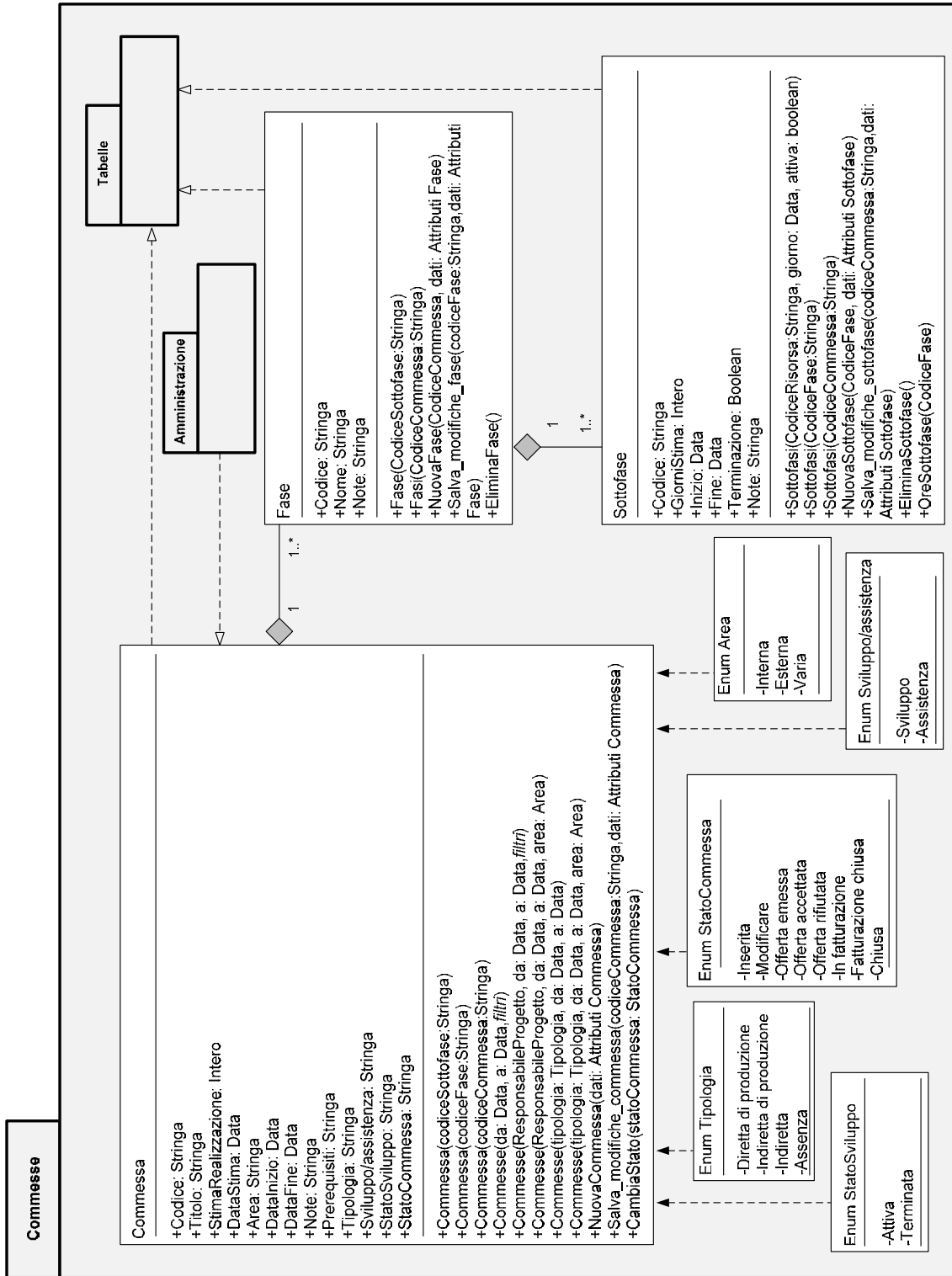


Figura 30: Class diagram – package commesse

4.3.3 Package amministrazione

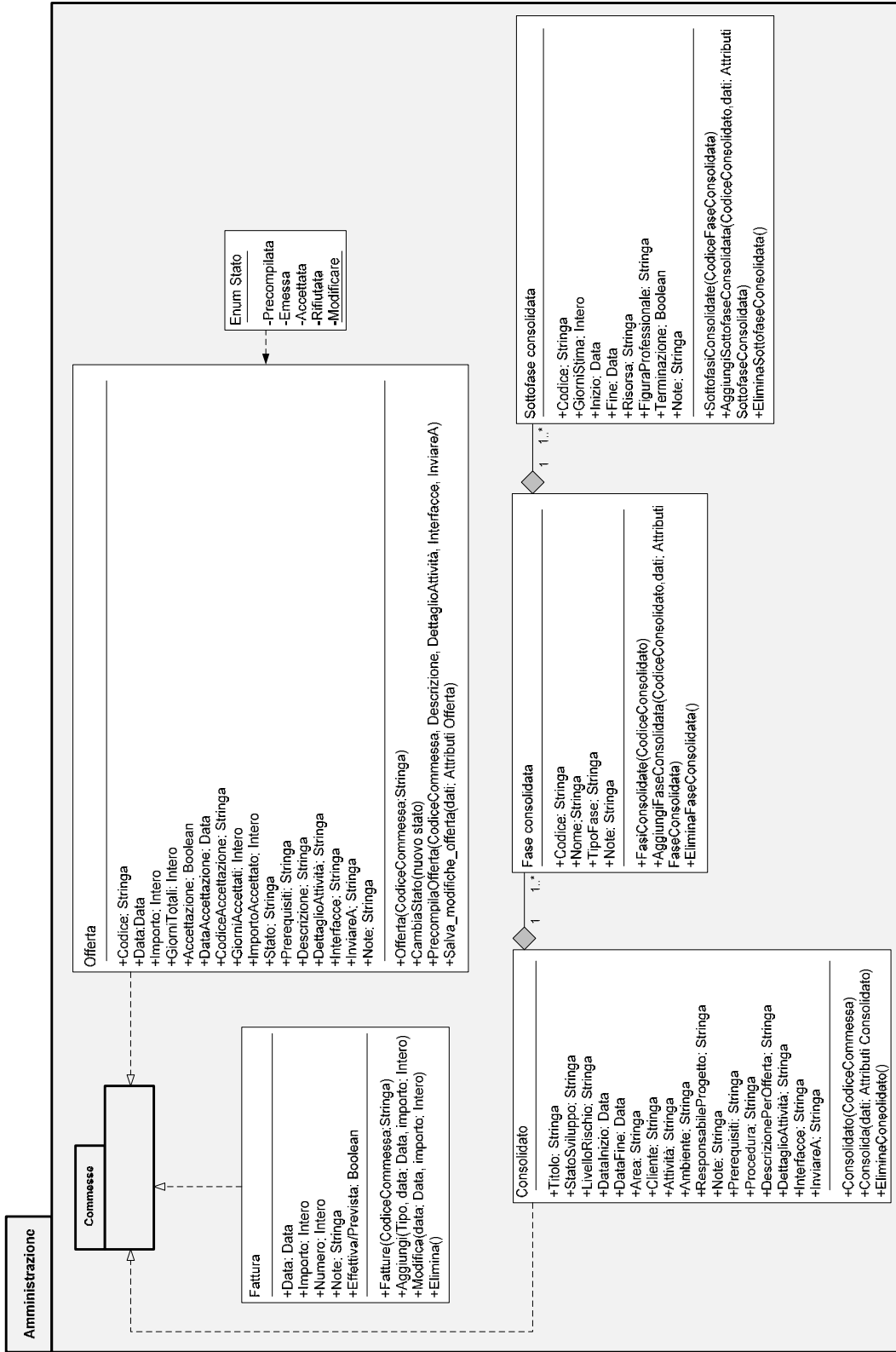


Figura 31: Class diagram – package amministrazione

4.3.4 Package tabelle

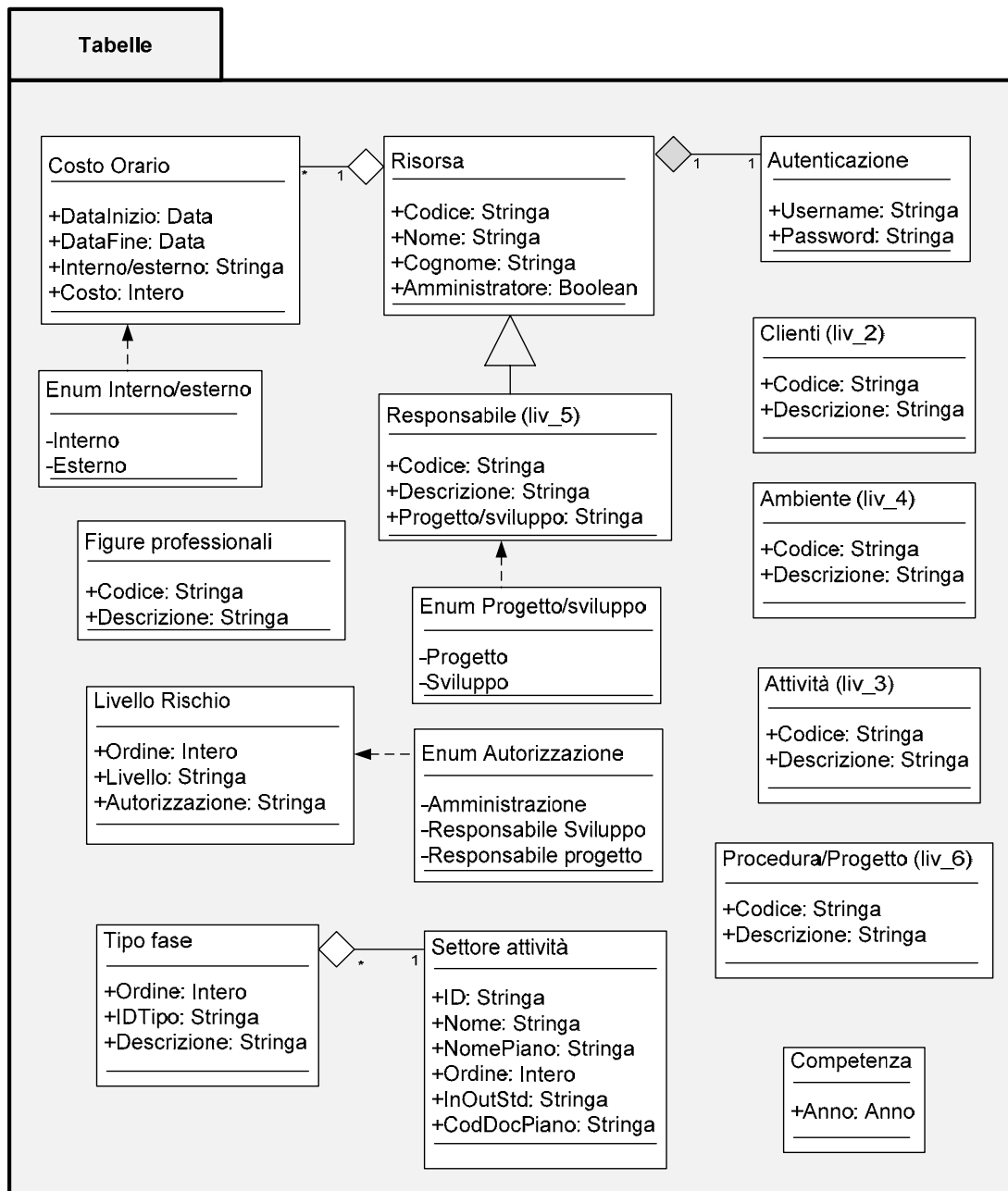


Figura 32: Class diagram – package tabelle, attributi

NOTE:

- Per le caratteristiche delle classi nel package tabelle i metodi consistono nell’inserimento, cancellazione, modifica e lettura, si omette quindi di elencarli.

4.3.5 Interazioni tra packages

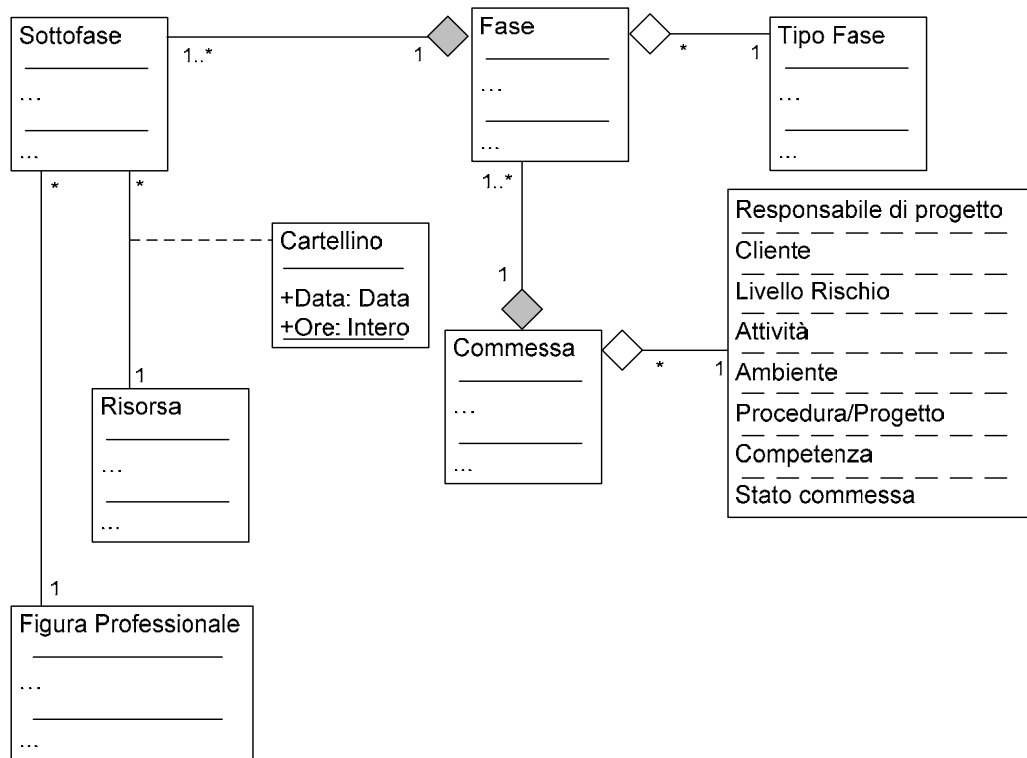


Figura 33: Class diagram – interazioni commesse/tabelle

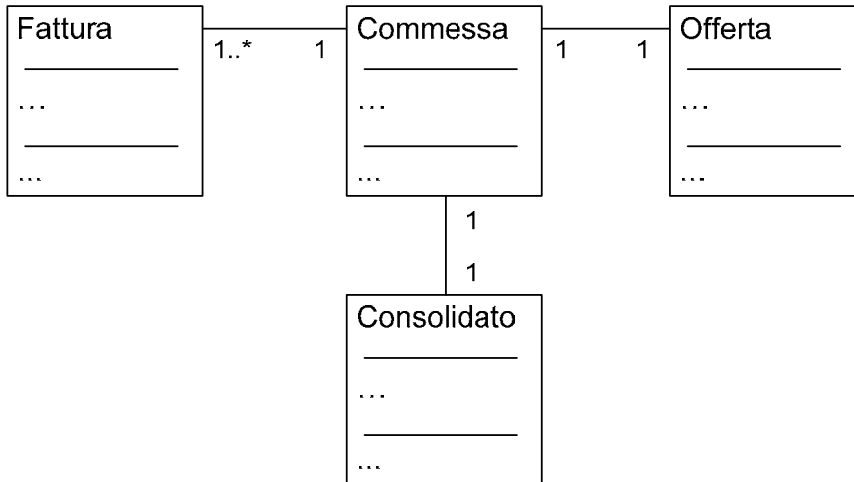


Figura 34: Class diagram – interazioni commesse/amministrazione

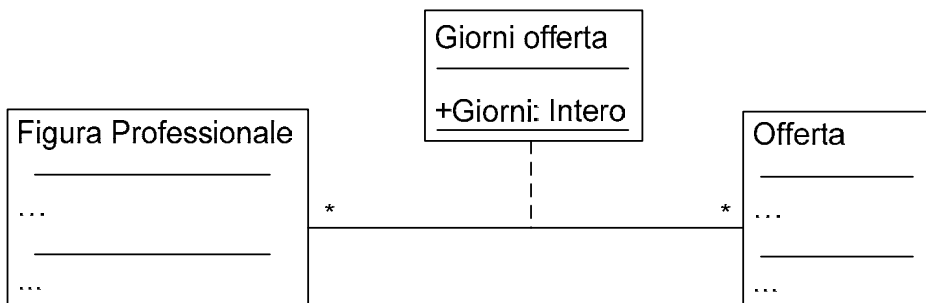


Figura 35: Class diagram – interazioni amministrazione/tabelle

4.4 Sequence diagram

Con i diagrammi di sequenza si possono modellare le interazioni tra le istanze delle classi introdotte precedentemente ed il sistema stesso evidenziandone l'ordine in maniera visuale.

4.4.1 Accesso

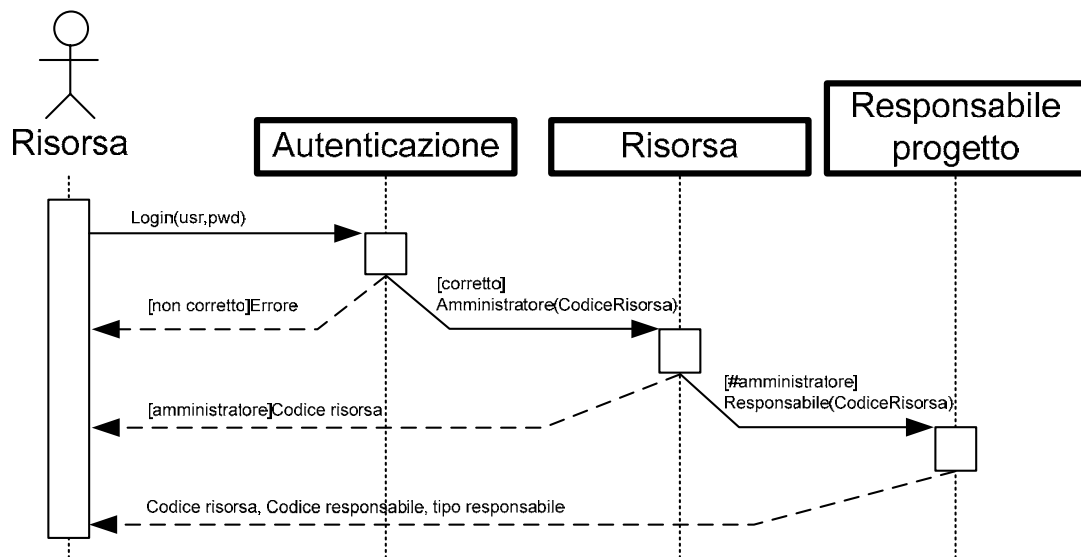


Figura 36: Sequence diagram – accesso

4.4.2 Cartellino

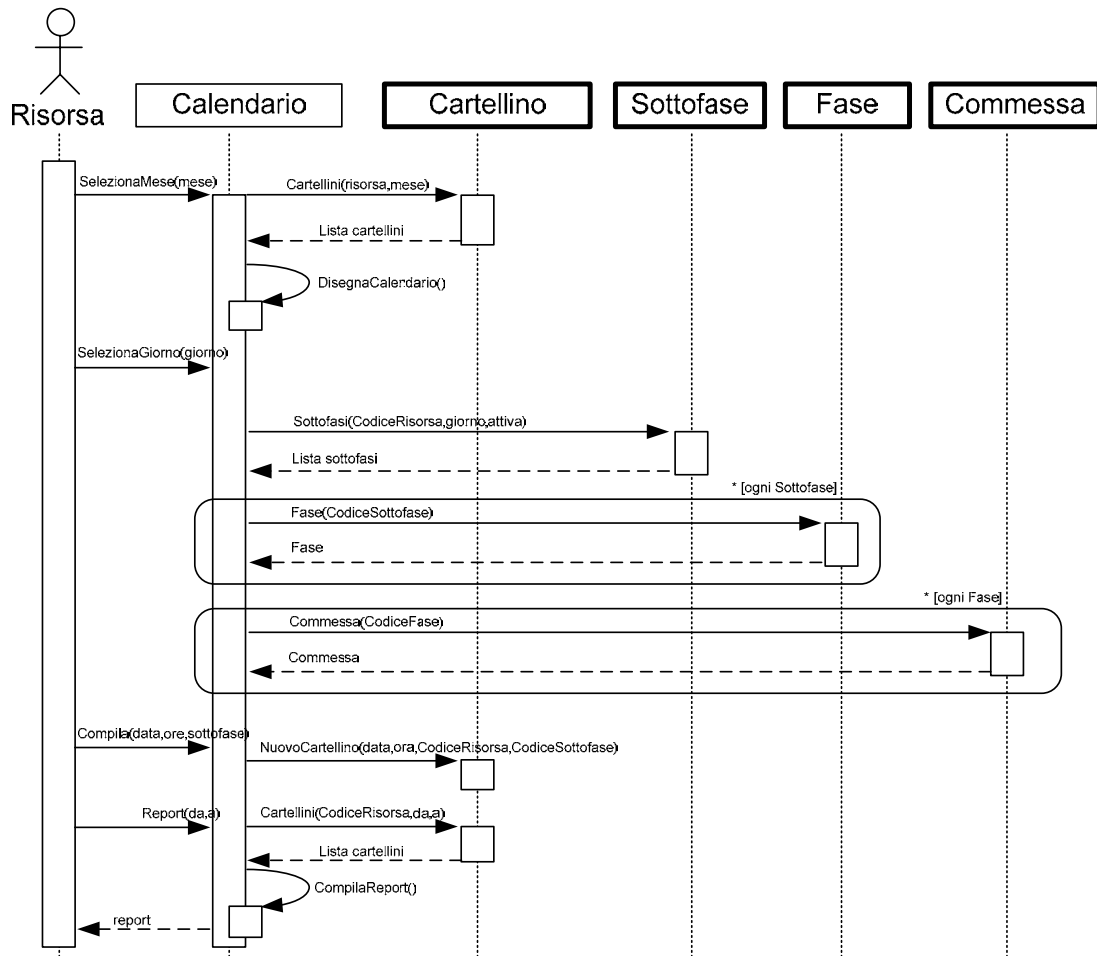


Figura 37: Sequence diagram – cartellino, risorsa

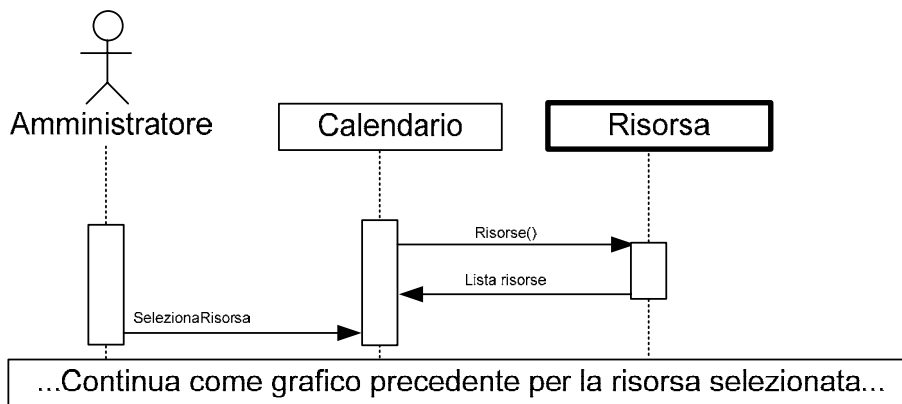


Figura 38: Sequence diagram – cartellino, amministratore

4.4.3 Produzione

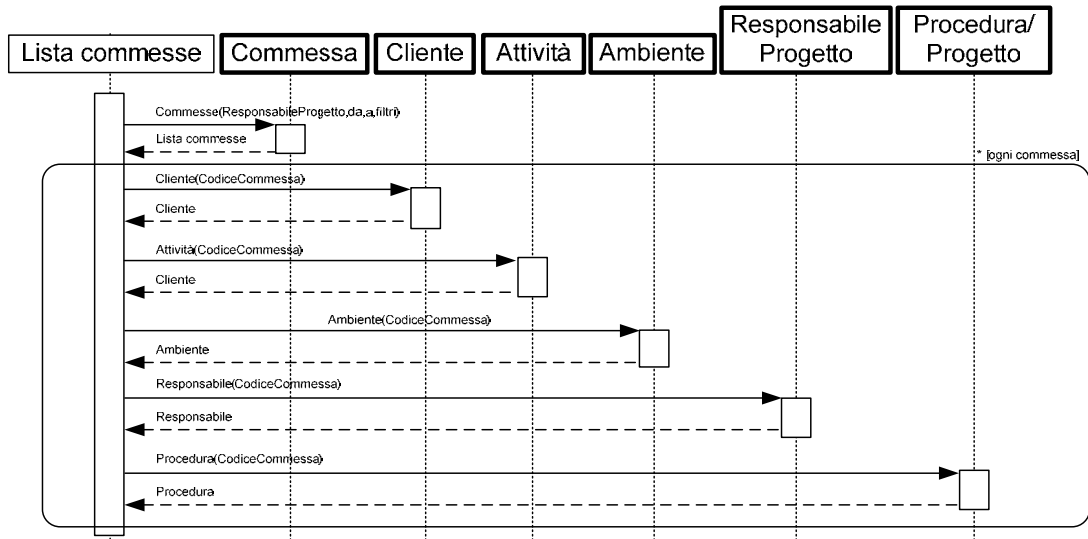


Figura 39: Sequence diagram – produzione, lista commesse

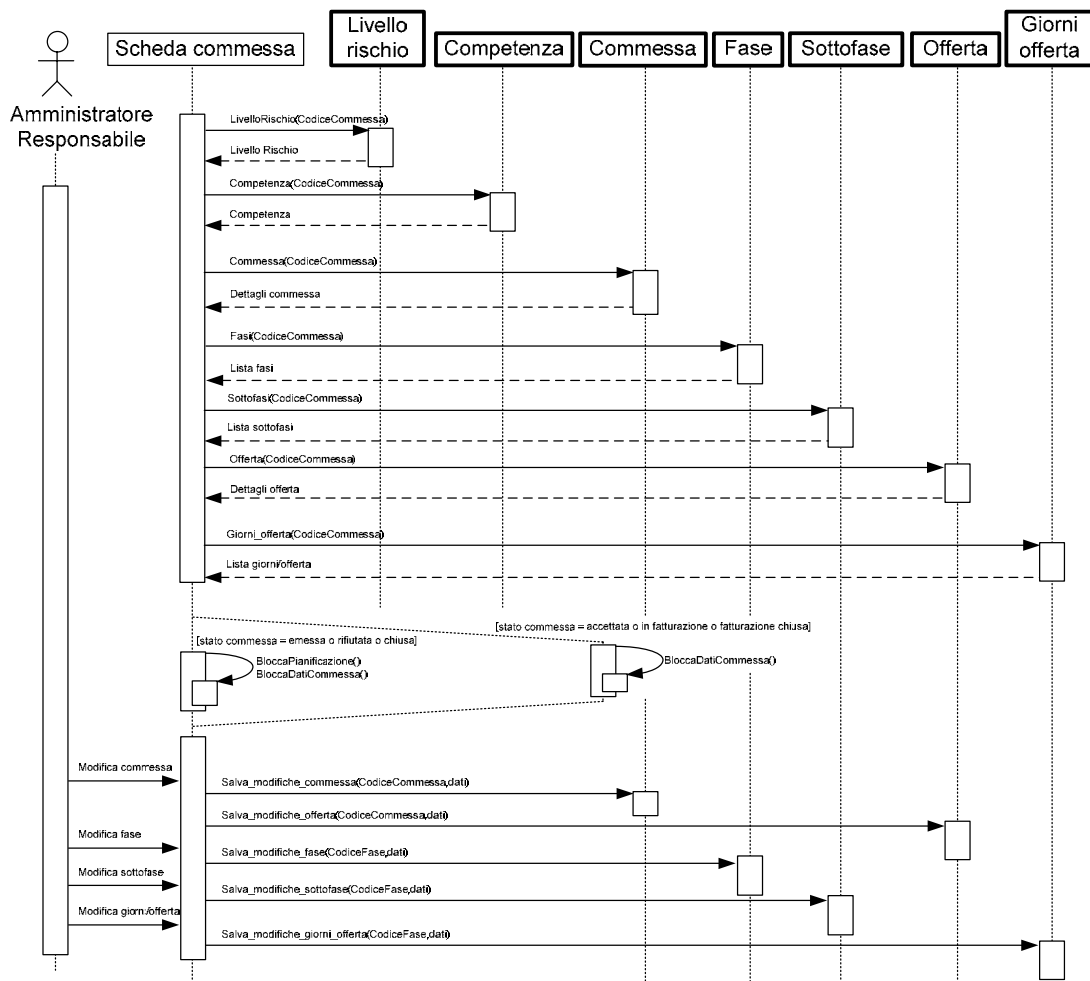


Figura 40: Sequence diagram – produzione, dettagli commessa

La visualizzazione della scheda commessa in modalità modificabile prevede che siano caricate le tabelle necessarie per la modifica, in *Figura 37* si è ommesso di esplicitare tale operazione ma ci si può riferire alla *Figura 38*.

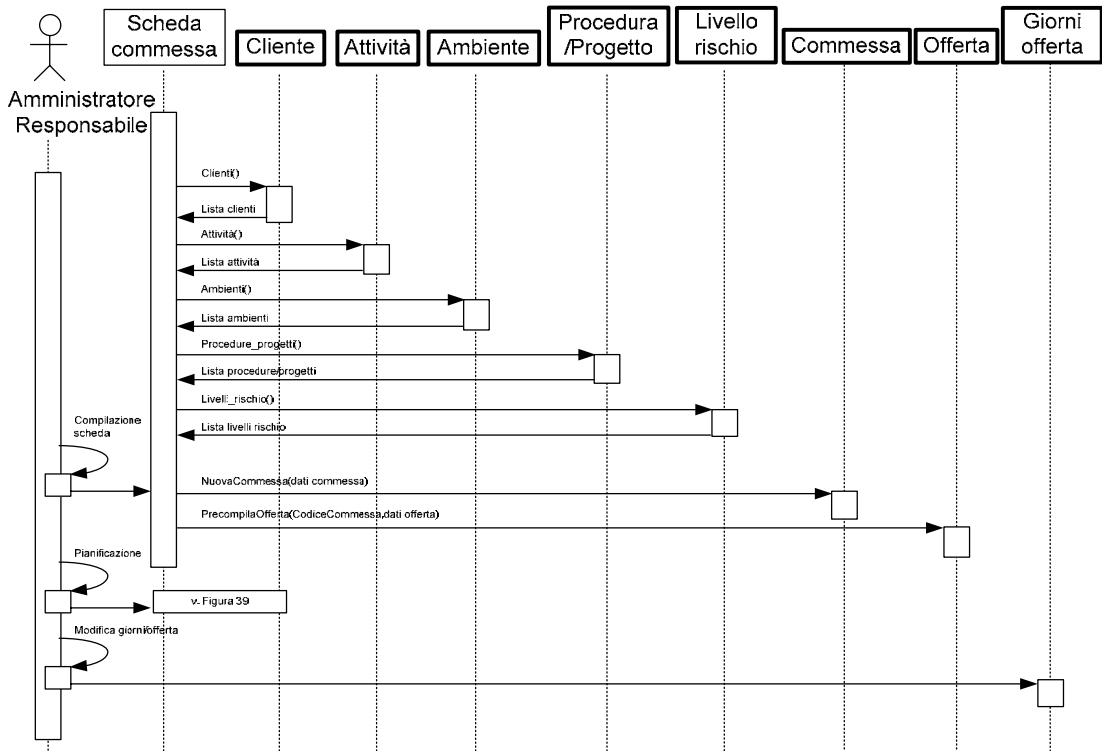


Figura 41: Sequence diagram – produzione, nuova commessa

NOTE:

- Per i dati commessa e dati offerta riferirsi a 2.6.11.2 *Requisiti d'interfaccia utente*.

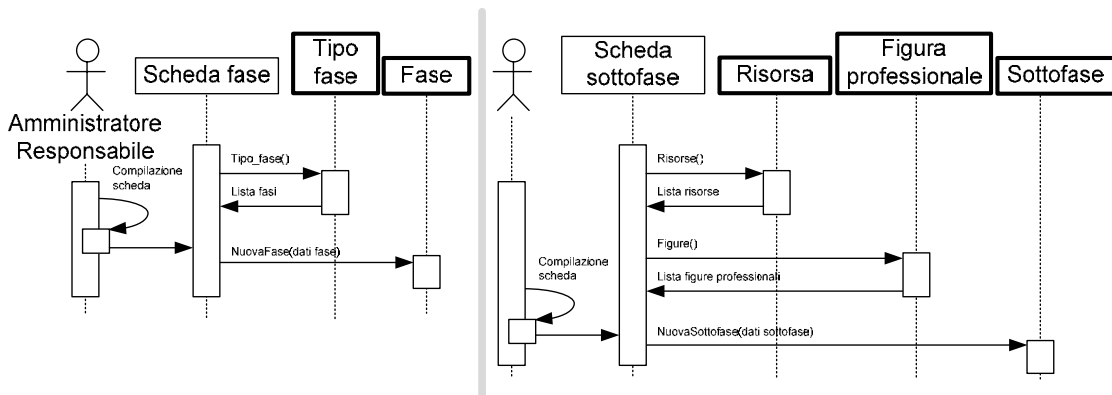


Figura 42: Sequence diagram – produzione, nuova fase e nuova sottofase

4.4.4 Amministrazione

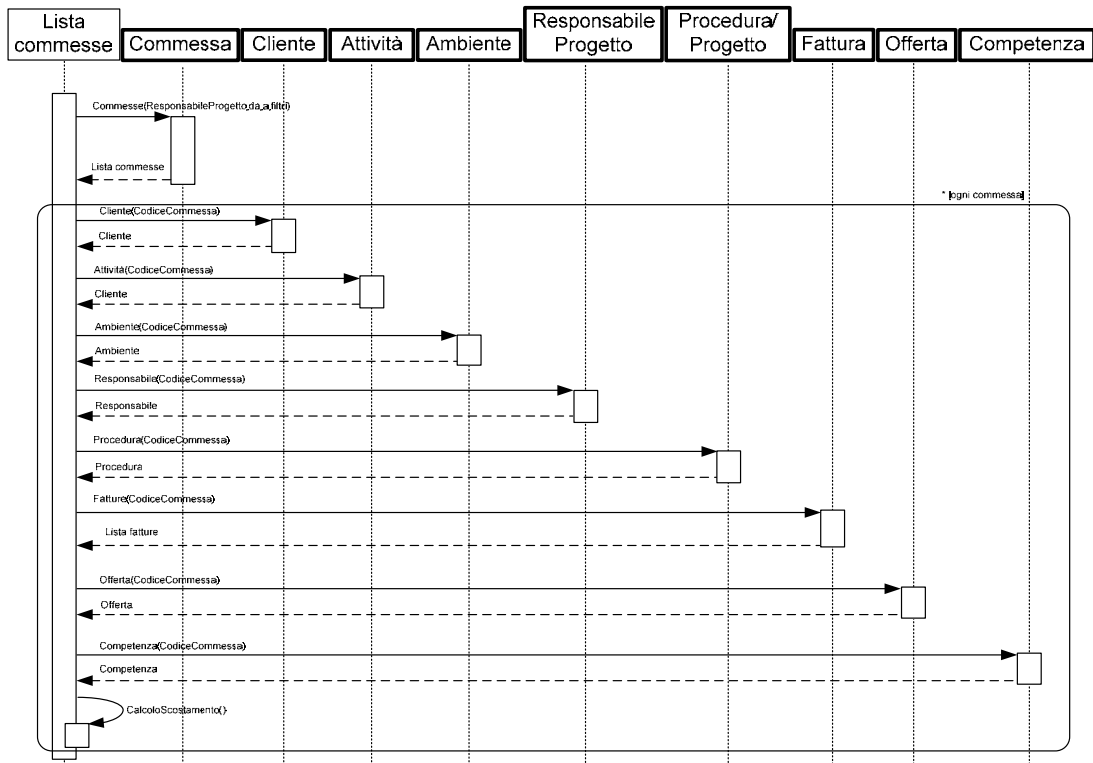


Figura 43: Sequence diagram – amministrazione, lista commesse

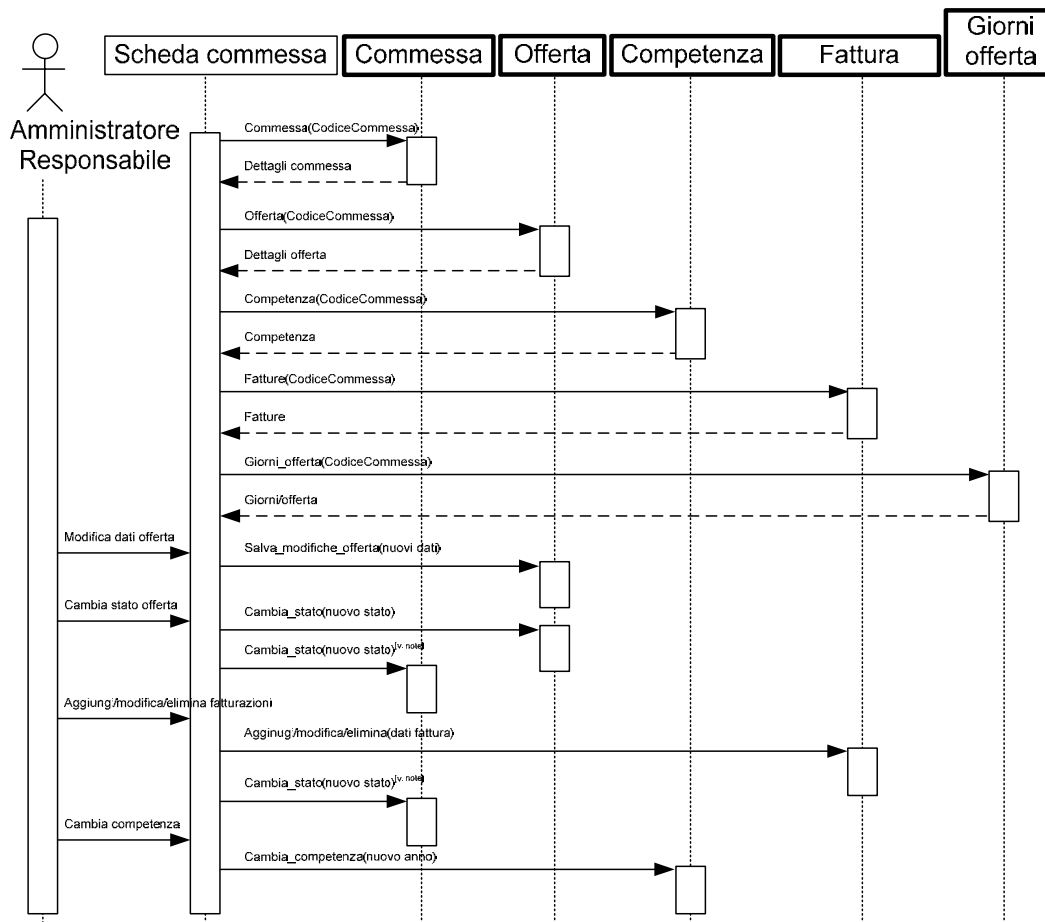


Figura 44: Sequence diagram – amministrazione, dettagli commessa

NOTE:

- I cambiamenti di stato delle commesse rispetto alle offerte e fatturazioni seguono le regole descritte in 2.6.6 *Stati commessa-offerta e interazioni*.

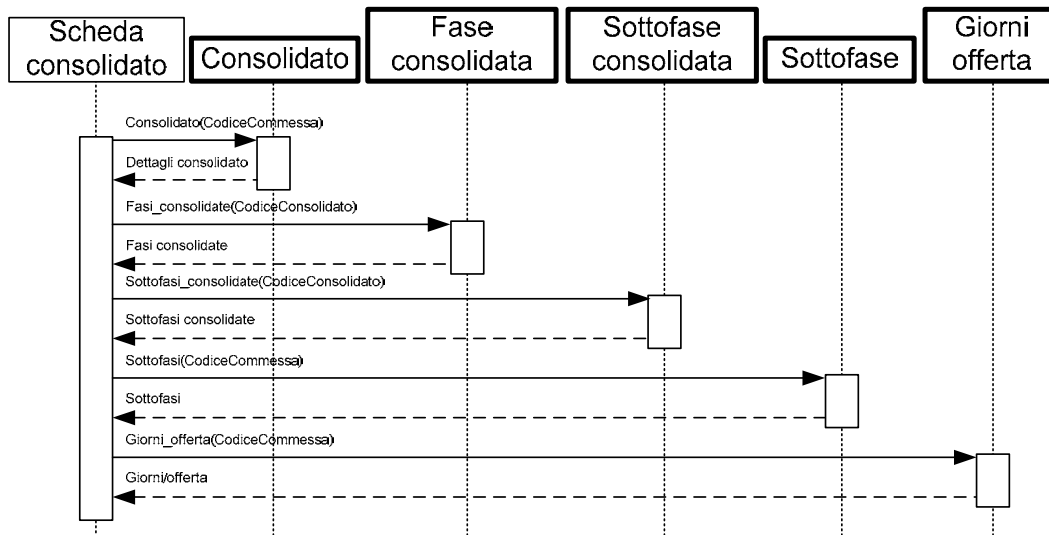


Figura 45: Sequence diagram – amministrazione, visualizza consolidato

NOTE:

- Le sottofasi vengono caricate per visualizzare un confronto tra i giorni accordati nell'offerta e i giorni attualmente in pianificazione.

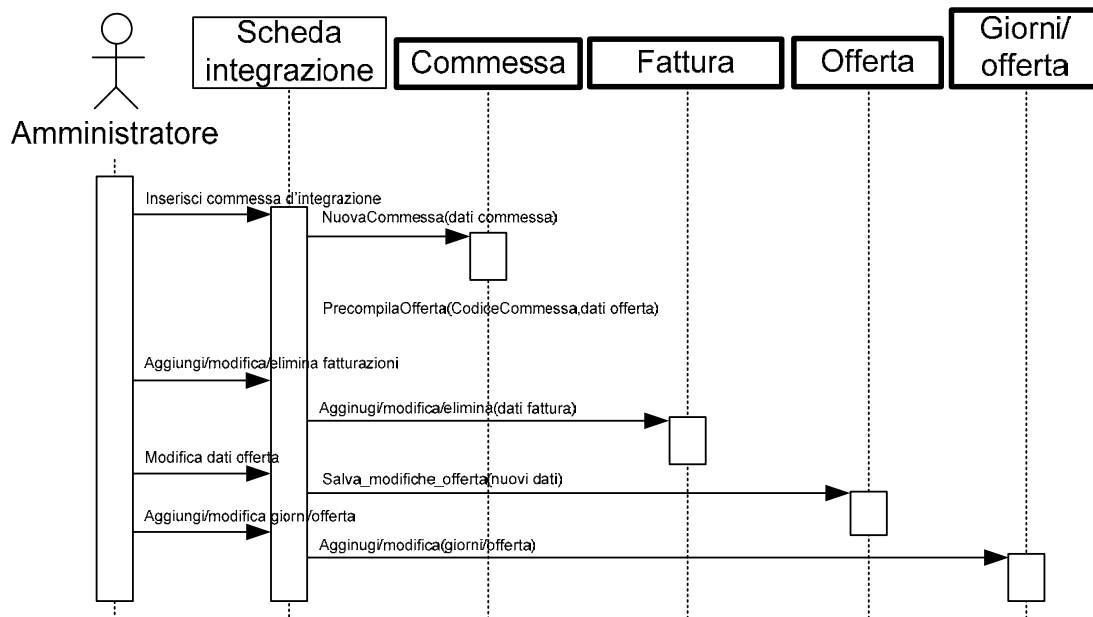


Figura 46: Sequence diagram – amministrazione, crea integrazione

NOTE:

- Per i dettagli riguardo la commessa d'integrazione riferirsi a 2.6.7 *Integrazione*.

4.4.5 Controllo

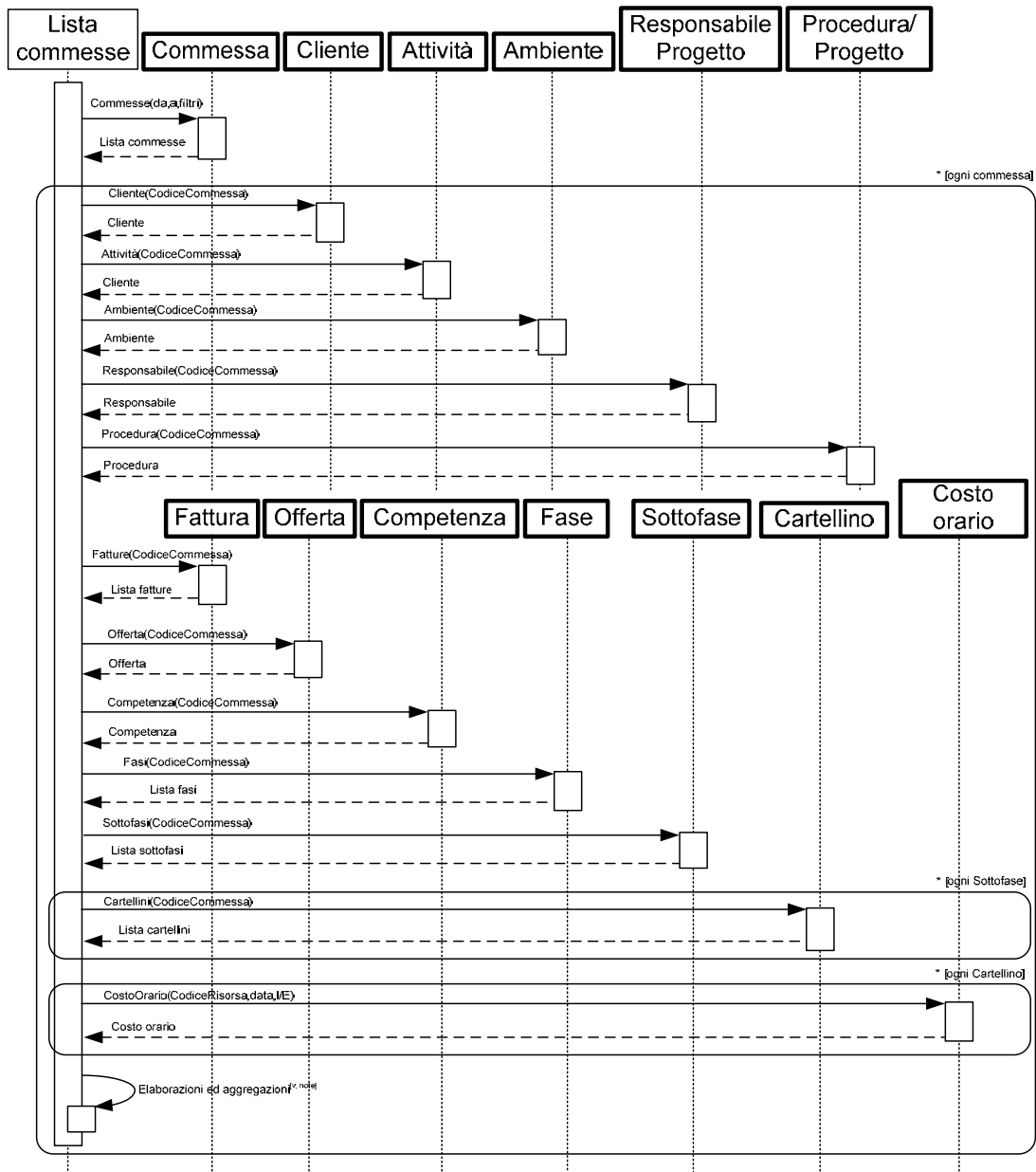


Figura 47: Sequence diagram – controllo, lista commesse

NOTE:

- Per i dettagli riguardo le elaborazioni, aggregazioni e filtri riferirsi a 3.5.1 *Lista commesse*.

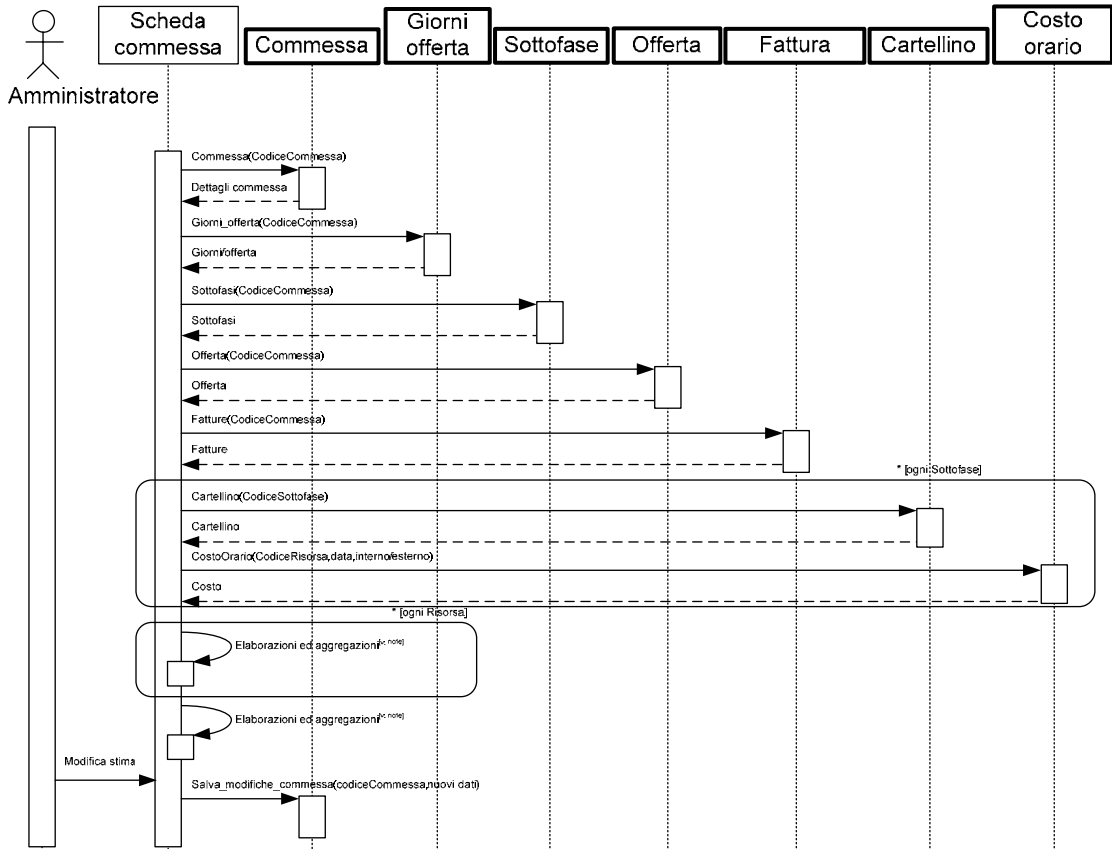


Figura 48: Sequence diagram – controllo, dettagli commessa

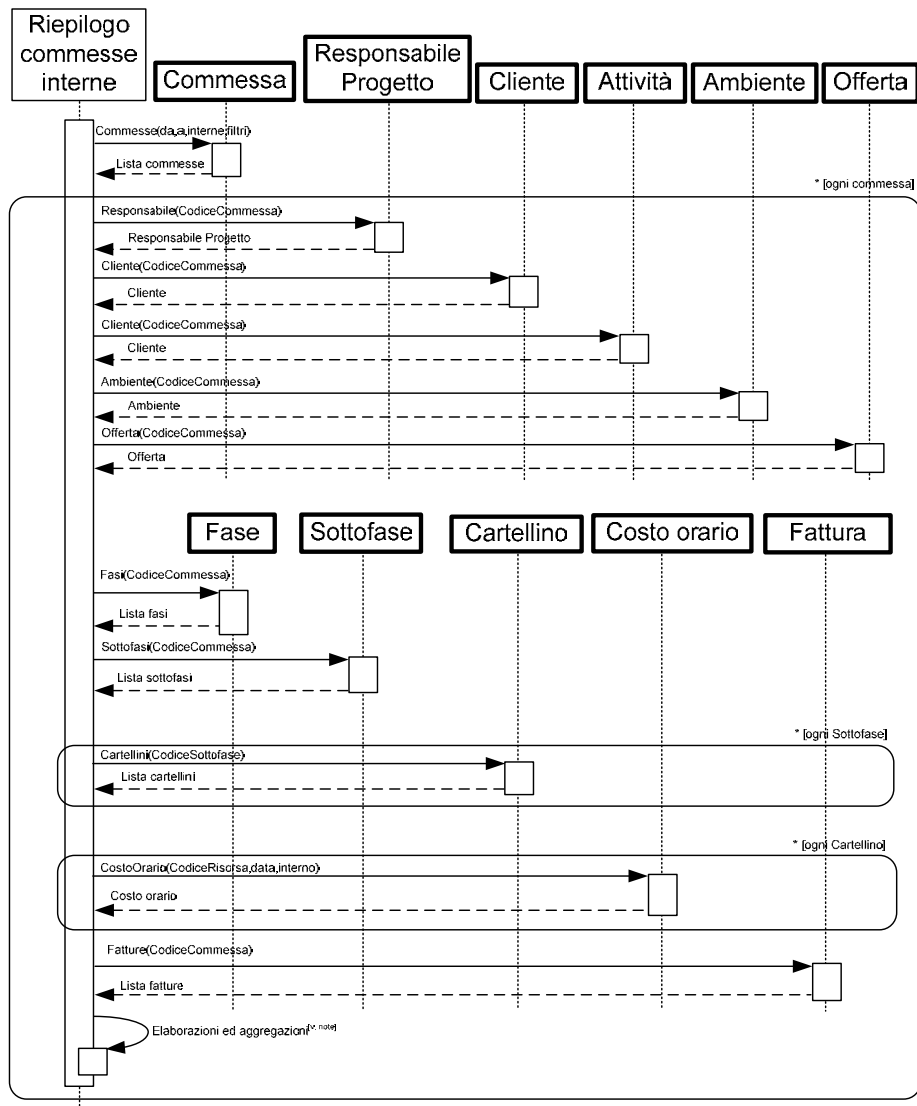


Figura 49: Sequence diagram – controllo, riepilogo commesse interne

NOTE:

- Per i dettagli riguardo le elaborazioni, aggregazioni e filtri riferirsi a 3.5.3 *Riepilogo commesse interne*.

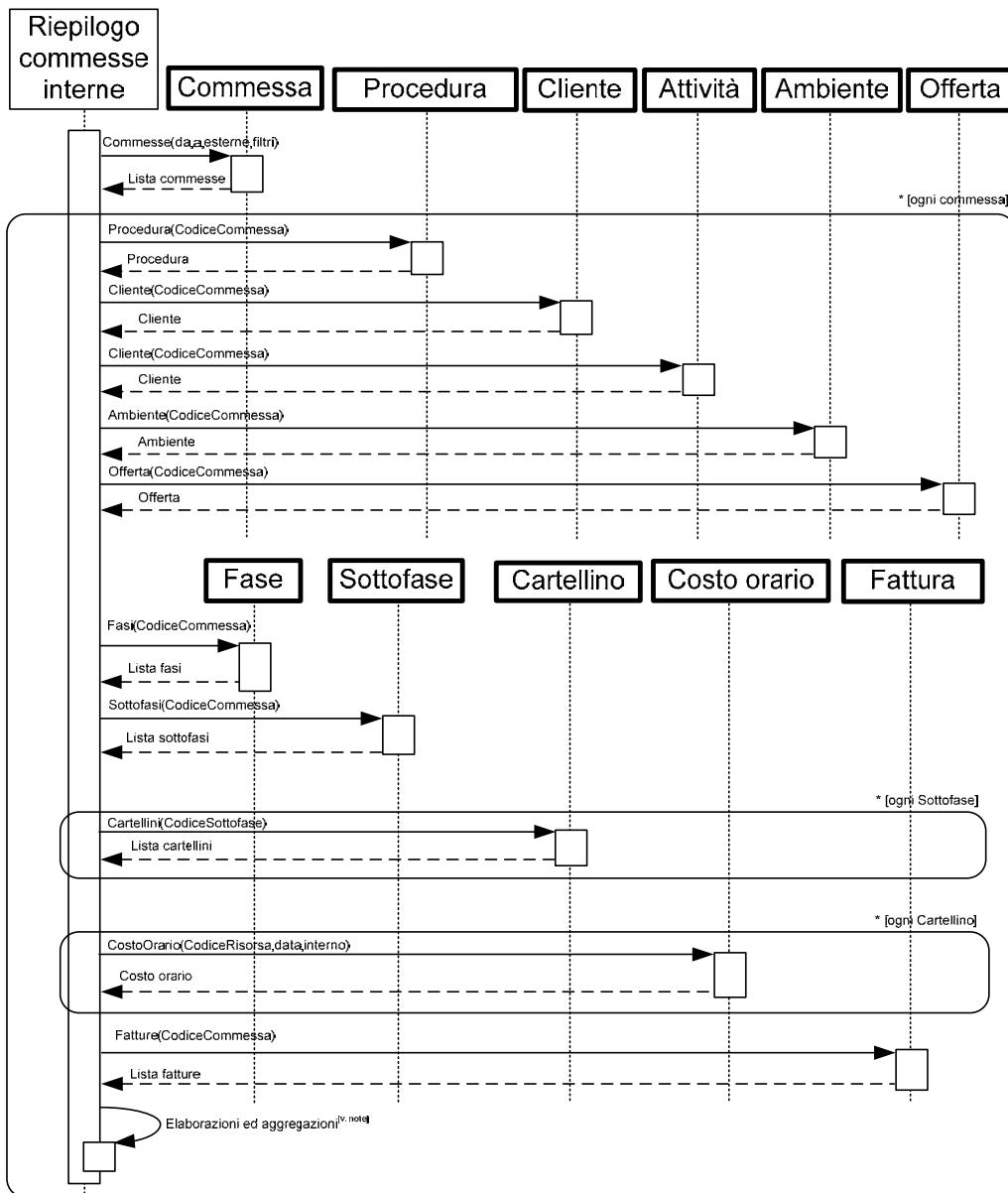


Figura 50: Sequence diagram – controllo, riepilogo commesse esterne

NOTE:

- Per i dettagli riguardo le elaborazioni, aggregazioni e filtri riferirsi a 3.5.4 *Riepilogo commesse esterne*.

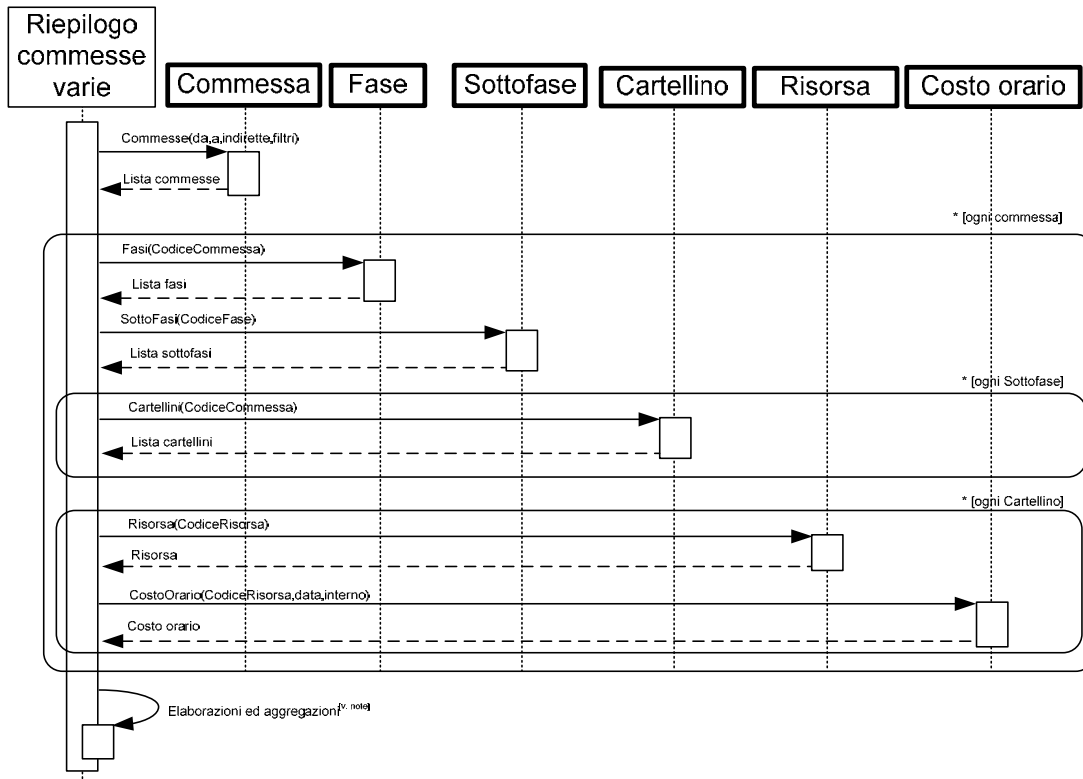


Figura 51: Sequence diagram – controllo, riepilogo commesse indirette

NOTE:

- Per i dettagli riguardo le elaborazioni, aggregazioni e filtri riferirsi a 3.5.5 *Riepilogo commesse indirette*.

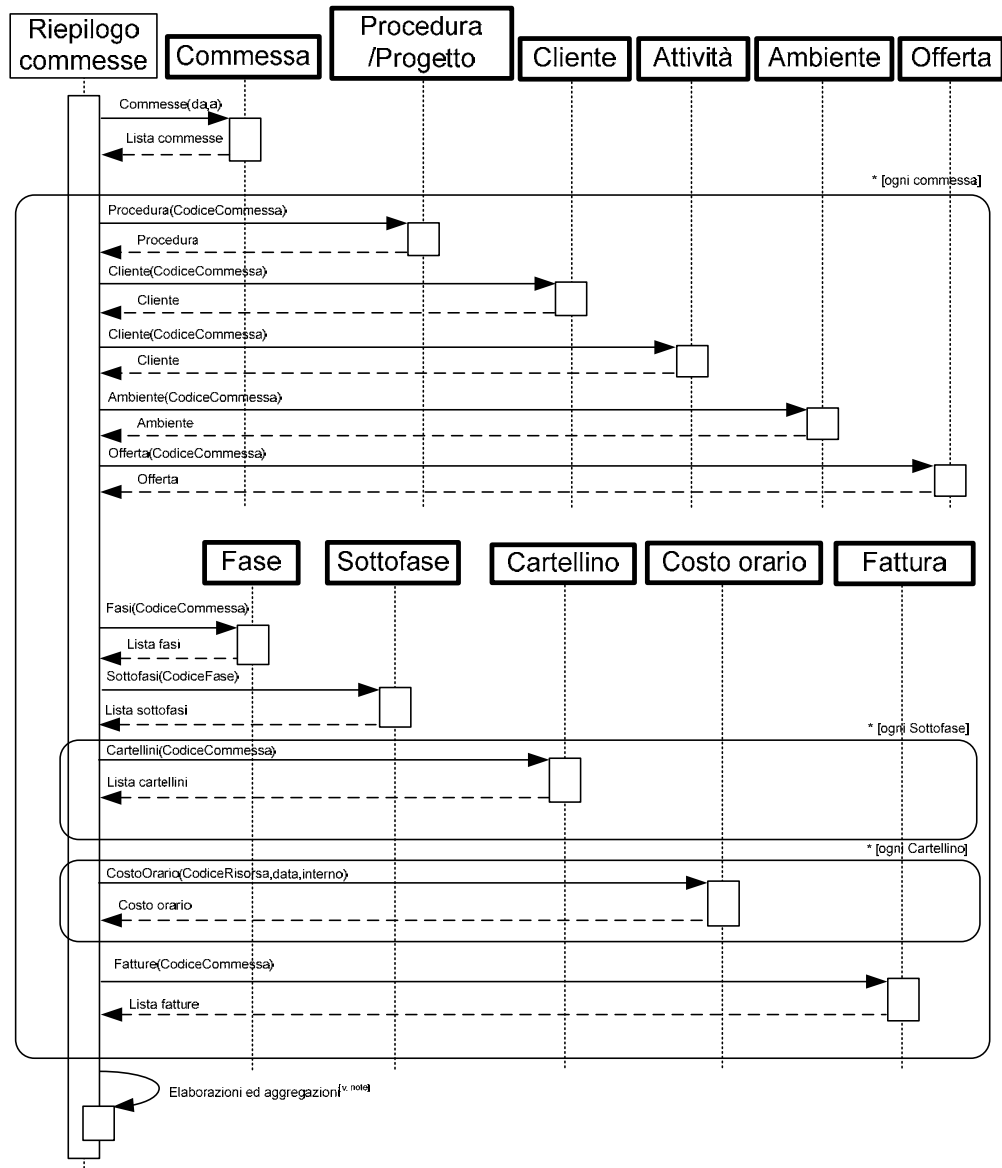


Figura 52: Sequence diagram – controllo, riepilogo commesse totale

NOTE:

- Per i dettagli riguardo le elaborazioni ed aggregazioni riferirsi a 3.5.6 *Riepilogo commesse totale*.

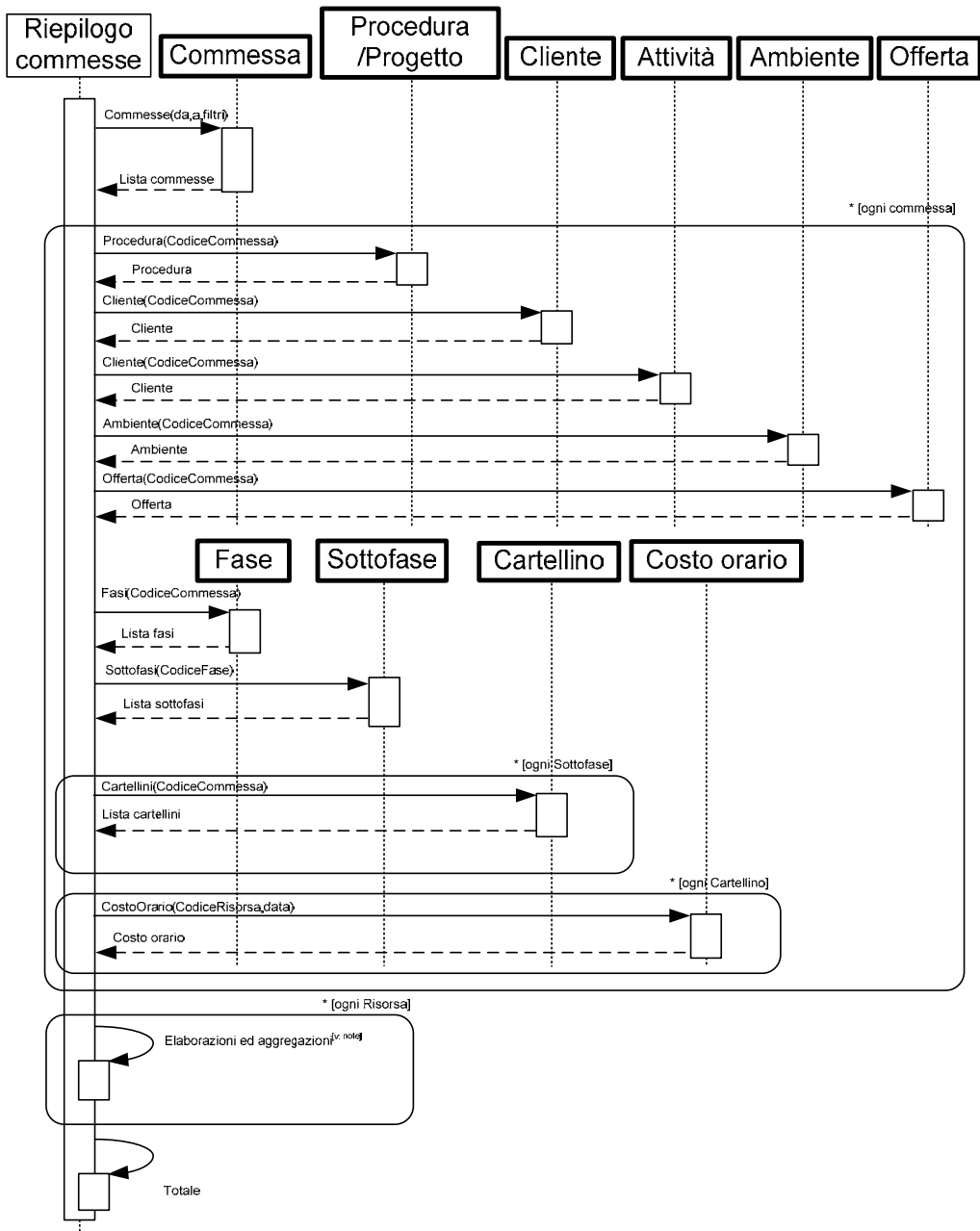


Figura 53: Sequence diagram – controllo, riepilogo risorse

NOTE:

- Per i dettagli riguardo le elaborazioni ed aggregazioni riferirsi a 3.5.7 *Riepilogo risorse*.

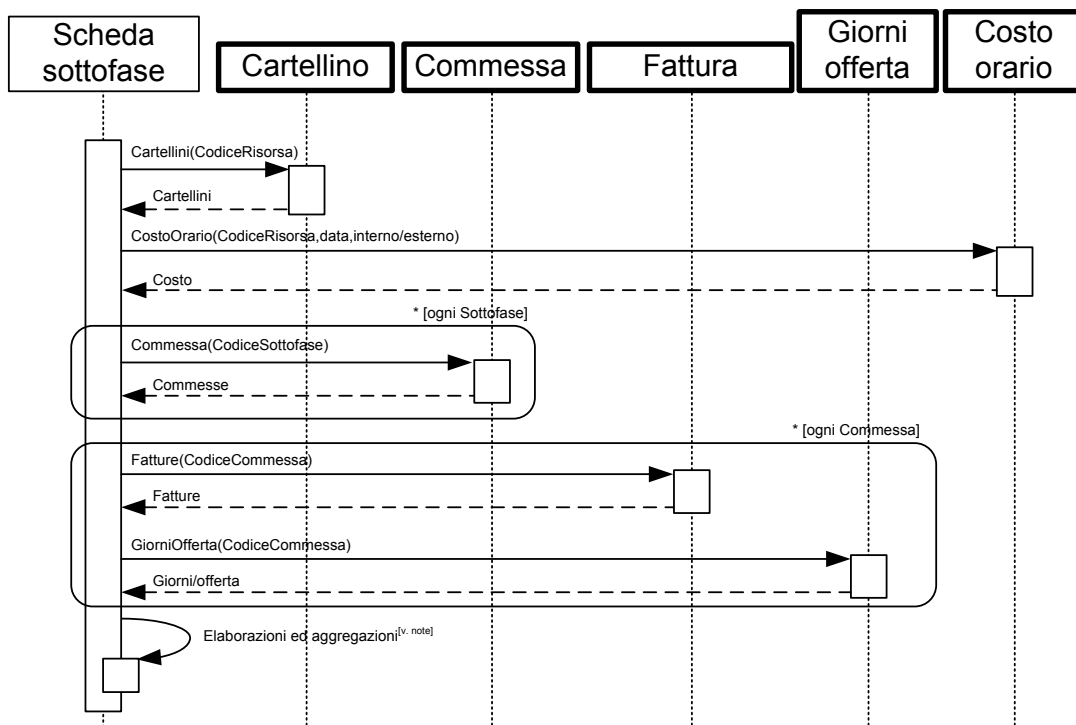


Figura 54: Sequence diagram – controllo, dettaglio risorsa

NOTE:

- Per i dettagli riguardo le elaborazioni ed aggregazioni riferirsi a 3.5.8 *Dettaglio risorsa*.

4.5 Progetto concettuale e logico della base di dati

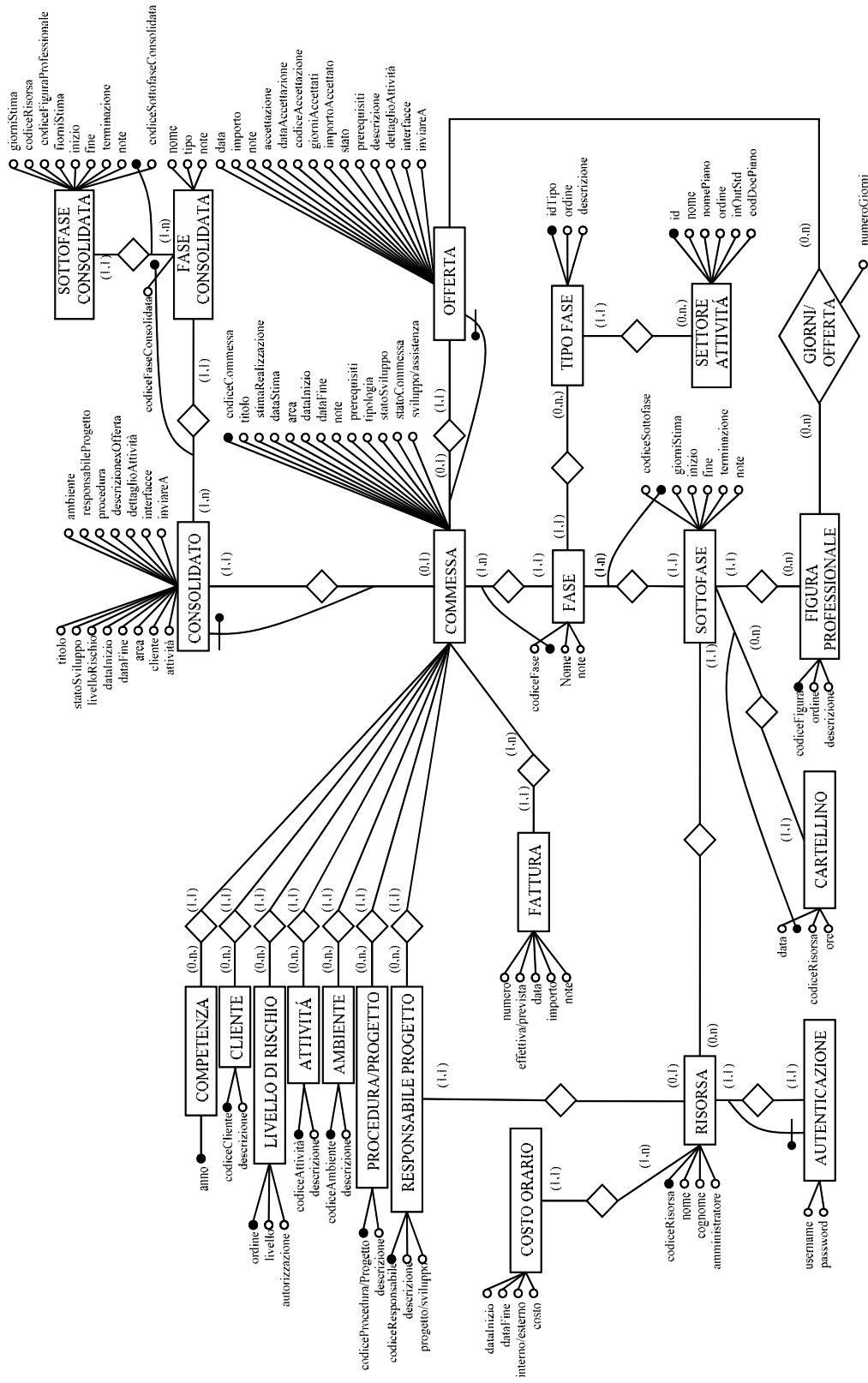


Figura 55: Schema E/R

4.5.1 Traduzione in schema relazionale

La traduzione dello schema E/R in tabelle relazionali è stato ottenuto attenendosi alle seguenti considerazioni:

- ogni entità, eccetto AUTENTICAZIONE, è tradotta con una relazione con gli stessi attributi e la stessa primary key;
- AUTENTICAZIONE può essere compattata in RISORSA in quanto l'associazione tra le due tabelle è unitaria e obbligatoria, la chiave primaria di AUTENTICAZIONE sarà quindi una chiave alternativa di RISORSA;
- tutte le associazioni, eccetto GIORNI/OFFERTA, presentano almeno un'entità che partecipa con cardinalità unitaria e verranno quindi compattate in tali entità includendo gli attributi dell'associazione e la chiave primaria dell'altra entità partecipante come foreign key;
- GIORNI/OFFERTA verrà tradotta con una tabella con gli stessi attributi più le primary key delle entità partecipanti come primary key, tali attributi si riferenziano alle rispettive tabelle.

Le tabelle risultanti sono:

COMMESSA	(<u>codiceCommessa</u> , titolo, stimaRealizzazione, dataStima, area, dataInizio, dataFine, note, prerequisiti, tipologia, sviluppo/assistenza, statoSviluppo, statoCommessa, codiceResponsabile, codiceCliente, livelloRischio, codiceAttività, codiceAmbiente, codiceProcedura/Progetto, annoCompetenza)
FK	(codiceResponsabile) references RESPONSABILE NOT NULL
FK	(codiceCliente) references CLIENTE NOT NULL
FK	(livelloRischio) references LIVELLO_RISCHIO NOT NULL
FK	(codiceAttività) references ATTIVITÀ NOT NULL
FK	(codiceAmbiente) references AMBIENTE NOT NULL
FK	(codiceProcedura/Progetto) references PROCEDURA/PROGETTO NOT NULL
FK	(annoCompetenza) references COMPETENZA NOT NULL
FASE	(<u>codiceCommessa</u> , <u>codiceFase</u> , nome, note, tipoFase)
FK	(codiceCommessa) references COMMESSA NOT NULL
FK	(tipoFase) references TIPO_FASE NOT NULL
SOTTOFASE	(<u>codiceCommessa</u> , <u>codiceFase</u> , <u>codiceSottofase</u> , codiceRisorsa, codiceFiguraProfessionale, giorniStima, inizio, fine, terminazione, note)
FK	(codiceCommessa, codiceFase) references FASE
FK	(codiceRisorsa) references RISORSA
FK	(codiceFiguraProfessionale) references FIGURA_PROFESIONALE

FATTURA FK	(<u>codiceCommessa</u> , <u>numero</u> , <u>effettiva/prevista</u> , data, importo, note) (codiceCommessa) references COMMESSA NOT NULL
OFFERTA FK	(<u>codiceCommessa</u> , <u>codice Accettazione</u> , data, importo, <u>dataAccettazione</u> , <u>giorniAccettati</u> , <u>importoAccettato</u> , stato, note, <u>prerequisiti</u> , <u>descrizione</u> , dettaglioAttività, interfacce, <u>inviareA</u>) (codiceCommessa) references COMMESSA NOT NULL
GIORNI OFFERTA FK FK	(<u>codiceCommessa</u> , <u>codiceFigura</u> , <u>numeroGiorni</u>) (codiceCommessa) references COMMESSA NOT NULL (codiceFigura) references FIGURA_PROFESSIONALE NOT NULL
CONSOLIDATO FK	(<u>codiceCommessa</u> , <u>titolo</u> , <u>statoSviluppo</u> , <u>livelloRischio</u> , <u>dataInizio</u> , <u>dataFine</u> , <u>codiceArea</u> , <u>codiceCliente</u> , <u>codiceAttività</u> , <u>codiceAmbiente</u> , <u>responsabileProgetto</u> , <u>codiceProcedura/Progetto</u> , note, <u>descrizione Offerta</u> , <u>prerequisiti</u> , dettaglioAttività, interfacce, <u>inviareA</u>) (codiceCommessa) references COMMESSA NOT NULL
FASE_CONSOLIDATA FK	(<u>codiceConsolidato</u> , <u>codiceFaseConsolidata</u> , <u>nome</u> , <u>tipo</u> , note) (codiceConsolidato) references CONSOLIDATO NOT NULL
SOTTOFASE_CONSOLIDATA FK	(<u>codiceConsolidato</u> , <u>codiceFaseConsolidata</u> , <u>codiceSottofaseConsolidata</u> , <u>codiceRisorsa</u> , <u>codiceFiguraProfessionale</u> , <u>giorni Stima</u> , <u>inizio</u> , <u>fine</u> , <u>terminazione</u> , note) (codiceConsolidato, codiceFaseConsolidata) references FASE_CONSOLIDATA NOT NULL
RISORSA AK	(<u>codiceRisorsa</u> , <u>nome</u> , <u>cognome</u> , <u>amministratore</u> , <u>username</u> , <u>password</u>) (username)
RESPONSABILE FK	(<u>codiceResponsabile</u> , <u>codiceRisorsa</u> , <u>descrizione</u> , <u>progetto/sviluppo</u>) (codiceRisorsa) references RISORSA NOT NULL
COSTO_ORARIO FK	(<u>codiceRisorsa</u> , <u>dataInizio</u> , <u>dataFine</u> , <u>interno/esterno</u> , <u>costo</u>) (codiceRisorsa) references RISORSA NOT NULL
CLIENTE	(<u>codiceCliente</u> , <u>descrizione</u>)
ATTIVITÀ	(<u>codiceAttività</u> , <u>descrizione</u>)
AMBIENTE	(<u>codiceAmbiente</u> , <u>descrizione</u>)
COMPETENZA	(<u>anno</u>)
LIVELLO_RISCHIO	(<u>ordine</u> , <u>livello</u> , <u>autorizzazione</u>)
FIGURA_PROFESSIONALE	(<u>codiceFigura</u> , <u>ordine</u> , <u>descrizione</u>)

PROCEDURA/PROGETTO

(codiceProcedura/Progetto, descrizione)

SETTORE_ATTIVITÀ

(idSettoreAttività, nome, nomePiano, ordine, inOutStd, codDocPiano)

TIPO_FASE

FK

(idTipo, idSettoreAttività, ordine, descrizione)

(idSettoreAttività) references SETTORE_ATTIVITÀ NOT NULL

CARTELLINO

FK

(codiceCommessa, codiceFase, codiceSottofase, data, codiceRisorsa, ore)

(codiceCommessa, codiceFase, codiceSottofase)

references SOTTOFASE NOT NULL

4.5.2 Trigger

Si è deciso di creare un gruppo di trigger per la transazione in automatico degli stati delle commesse in base alla rispettiva offerta e alle fatturazioni come illustrato in *2.6.6 Stati commessa-offerta e fatturazioni*.

TABELLA	EVENTO	CONDIZIONE	AZIONE	
Commessa	creazione		inserimento Offerta in stato "precompilata"	1
Offerta	cambiamento stato	nuovo stato = "emessa"	stato Commessa = "offerta emessa"	2
			registrazione Consolidato	
		nuovo stato = "accettata"	stato Commessa = "offerta accettata"	3
		nuovo stato = "rifiutata"	stato Commessa = "offerta rifiutata"	4
			stato Sviluppo = "terminato"	
		nuovo stato = "modificare"	stato Commessa = "modificare"	5
			eliminazione Consolidato	
Fattura	inserimento	somma prevista # somma effettiva	stato commessa = "in fatturazione"	6
		somma prevista = somma effettiva	stato Commessa = "fatturazione chiusa"	7

Figura 56: Trigger

5. Sviluppo di Syndev

Dopo aver determinato i requisiti che Syndev dovrà soddisfare e aver prodotto un progetto UML si è passati alla fase di sviluppo del software.

5.1 Intranet e applicazioni web

L'azienda ospitante utilizza una *Intranet* con visibilità solo interna. Una *Intranet* aziendale facilita la condivisione di informazioni all'interno di organizzazioni in riservatezza, inoltre con il termine Intranet si intende non solo l'infrastruttura per la comunicazione tra i nodi ma anche l'insieme dei servizi forniti agli utenti dell'organizzazione. L'utilizzo di protocolli e tecnologie affermate in *Internet* e *World Wide Web* fornisce notevoli vantaggi^[37]:

- facilità di realizzazione con tecniche standard;
- bassi costi di implementazione;
- indipendenza dall'hardware;
- eventuale espansione di servizi con l'accesso a Internet;
- sicurezza e riservatezza, ad esempio con l'utilizzo di firewall e crittografia.

Alla luce di queste considerazioni si è deciso di sviluppare Syndev sotto forma di *applicazione web* accessibile all'interno dell'*Intranet* dell'azienda ospitante in quanto non è richiesta una visibilità esterna. Un'*applicazione web* è un'applicazione accessibile tramite un browser web, dal lato utente non è quindi distinguibile da pagine Web statiche. Nel tempo sono state proposte varie soluzioni per realizzare pagine web dinamiche, tra le più affermate vi sono^[38]:

- **CGI**: interfacce per l'esecuzione di applicazioni lato server esterne al server web, possono essere scritte in numerosi linguaggi ma presentano dei problemi di sicurezza in quanto permette l'esecuzione di programmi direttamente sulla macchina server;
- **Applet**: applicazioni scritte in java eseguite da lato client;

- **DHTML**: insieme di tecnologie per creare pagine web dinamiche^[39]:
 - *Javascript*: analogo al VBScript della Microsoft, script inseriti direttamente nella pagina web eseguiti dal browser, permettono di agire direttamente sugli oggetti che compongono la pagina secondo il modello DOM (*Document Object Model*);
 - *DOM*: definisce un set standard di oggetti HTML e il modo in cui manipolarli;
 - *CSS*: permette al web designer di definire dei *fogli di stile*, cioè la descrizione delle caratteristiche di oggetti *DOM*, da associare a una o più pagine web, permettendo così di modificarle tutte semplicemente agendo su un unico *foglio di stile*;
- **Servlet**: applicazioni in Java eseguite in un container di applicazioni sulla macchina server;
- **Linguaggi di scripting**: script inseriti direttamente nelle pagine HTML, il codice viene riconosciuto perché delimitato da tag ben definiti e interpretato generando così pagine a run-time anche in base alle richieste dell'utente, forniscono inoltre funzioni per l'accesso a database. Attualmente i linguaggi di scripting più diffusi sono:
 - *ASP*: Active Server Pages della Microsoft;
 - *PHP*: Hypertext Preprocessor, open source;
 - *JSP*: JavaServer Pages, della Sun;
- **Flash**: programma che consente di creare "movies" interattivi pubblicabili sul web, sfrutta la grafica vettoriale ed è facile da apprendere;
- **AJAX**: non un vero e proprio linguaggio di programmazione ma una tecnica per creare applicazioni web interattive veloci utilizzando una comunicazione tra un *javascript* e il server con trasferimento asincrono di dati, in questo modo le pagine web possono essere aggiornate del server tramite l'invio di un'esigua quantità di dati.

Il discorso riguardante le tecnologie web è trattato ampiamente nelle tesi pubblicate su *dbgroup.unimo.it*.

5.2 Framework di sviluppo

Nell'*Intranet* dell'azienda sono presenti diversi *server web*, tra questi sono stati messi a disposizione un DBMS *Microsoft SQL Server 2005*^[46] e più server con sistemi operativi *Windows* o *Linux*. Si è quindi scelto di realizzare l'applicazione web in PHP utilizzando le tecnologie *DHTML* e la base di dati sul DBMS *Microsoft SQL Server*.

Durante lo sviluppo è stata utilizzata in locale la piattaforma di sviluppo *Xampp 1.66a* per Windows, fa parte di un progetto open source e contiene un'insieme di applicazioni preinstallate ma comunque configurabili secondo le esigenze^[40]:

- **Apache HTTPD 2.2.8**: web server sviluppato da *Apache Software Foundation*;
- **Openssl 0.9.8g**: librerie di per l'esecuzione di funzioni crittografiche, implementano i protocolli SSL e TLS;
- **MySQL 5.0.51a**: DBMS relazionale sviluppato da *Sun Microsystems* con licenza open source *GNU GPL* o commerciale a seconda della versione e funzionalità fornite;
- **PHP 5.2.5**: ultima versione dell'interprete script PHP, è installato come modulo di *Apache*;
- **PHP 4.4.8**: versione precedente a *PHP 5.2.5*;
- **phpMyAdmin 2.11.4**: applicazione web sviluppata in PHP per la gestione semplificata di database su MySQL;
- **FileZilla FTP Server 0.9.25**: server FTP;
- **Mercury Mail Transport System 4.52**: mail server.

SyndeV è ancora in fase di sviluppo su piattaforma *XAMPP* ma, come è spiegato nell'ultimo capoverso di *4.a Database SyndeV*, non necessita di pesanti modifiche nel momento in cui si decida di utilizzare un altro server web con interprete *PHP*.

5.3 Architettura di *Syndev*

L'architettura di un'applicazione web segue tipicamente il classico approccio client/server in cui il client, tramite una richiesta, usufruisce dei servizi forniti server, in particolare il client è il browser dell'utente per l'accesso ai servizi forniti dall'applicazione web come ad esempio l'interfacciamento al database.

Dal lato server tipicamente un'applicazione web si sviluppa su tre livelli logico-funzionali che possono corrispondere a uno o più processi server su una o più macchine server:

- *presentazione*: logica di presentazione dei dati al client;
- *applicazione(o business)*: logica di elaborazione dei dati relativamente al servizio da offrire;
- *dati*: gestione dei dati, compreso l'accesso al database.

Il modello architetturale che è stato inizialmente preso in considerazione per lo sviluppo è il pattern *MVC* ^[41]:

- *Model*: rappresenta il livello dati e quindi le modalità di accesso e aggiornamento;
- *View*: gestisce la logica di presentazione dei dati;
- *Controller*: implementa una logica di controllo dell'applicazione traducendo le interazioni tra utente e il *view* in sequenza di azioni che il model deve svolgere;

Non è stata utilizzato il pattern *MVC* esattamente come descritto in letteratura^[41] ma è stata elaborata un'architettura ad hoc. Si è quindi deciso di organizzare i file contenenti gli script PHP di *Syndev* in tre directory omonime alle categorie di appartenenza delle funzioni svolte più una aggiuntiva di utility.

In particolare:

Model

Gli script appartenenti a questa categoria implementano le funzionalità di accesso alla base di dati, quindi è in questi file che avviene l'accesso al DBMS e le interrogazioni al database secondo le richieste degli script *controller*, i dati sono poi ritornati al *controller* che ha fatto la richiesta. Il *Model* è l'implementazione in PHP delle classi illustrate nel paragrafo 3.c *Class diagram*, infatti PHP fornisce i costrutti necessari per la programmazione a oggetti^[44], compaiono quindi i metodi descritti nei class diagram e altri metodi inseriti per facilitare lo sviluppo;

View

Implementa la logica di presentazione, gli script appartenenti a questa categoria ricevono i dati da presentare dal *controller* e gli elabora per generare pagine *DHTML* complete di form per interagire con l'utente e inviare gli input al *controller* necessario per rispondere alla richiesta. Gli script PHP del *view* sono organizzati in funzioni invocabili dall'esterno e raggruppati in file a seconda del contesto, infatti è presente un file per ogni frame che compone le pagine web. Inoltre nella stessa directory sono definiti i *fogli di stile* in file *.css*;

Controller

Svolge una funzione di arbitraggio e coordinamento delle interazioni tra *Model* e *View*, prende in input le richieste dell'utente, utilizza i metodi forniti dal *model* per aggiornare o interrogare la base di dati e fornisce i dati di ritorno alle funzioni di *view*. Inoltre si occupa di mantenere informazioni riguardo la sessione di lavoro settando dei "cookie" sul client, cioè informazioni che il server memorizza sul client il quale le invia ad ogni richiesta. La sequenza di operazioni da svolgere per soddisfare le richieste dell'utente viene decisa in base a un parametro dedicato tra gli input del form come se fosse ad esempio la chiamata di una funzione. Gli script *controller* sono riuniti in base al contesto, infatti sono presenti dei file che gestiscono tutte le richieste che vengono fatte dall'utente di fronte ad una lista di commesse piuttosto che al calendario per la compilazione del cartellino.

Tool

La directory *tool* contiene delle classi di supporto allo svolgimento delle altre funzioni, tali classi forniscono funzioni di:

- *connessione*: sono definiti i parametri di connessione alla base di dati, si può ottenere la connessione tramite la chiamata di un metodo, quindi in caso di modifica dei parametri di connessione va modificata solo una classe;
- *elaborazione stringhe data*: durante l'uso del programma emerge la necessità di formattare le stringhe corrispondenti alle date in diversi modi, ad esempio in base al formato *datetime* di *SQL Server* della *Microsoft*.

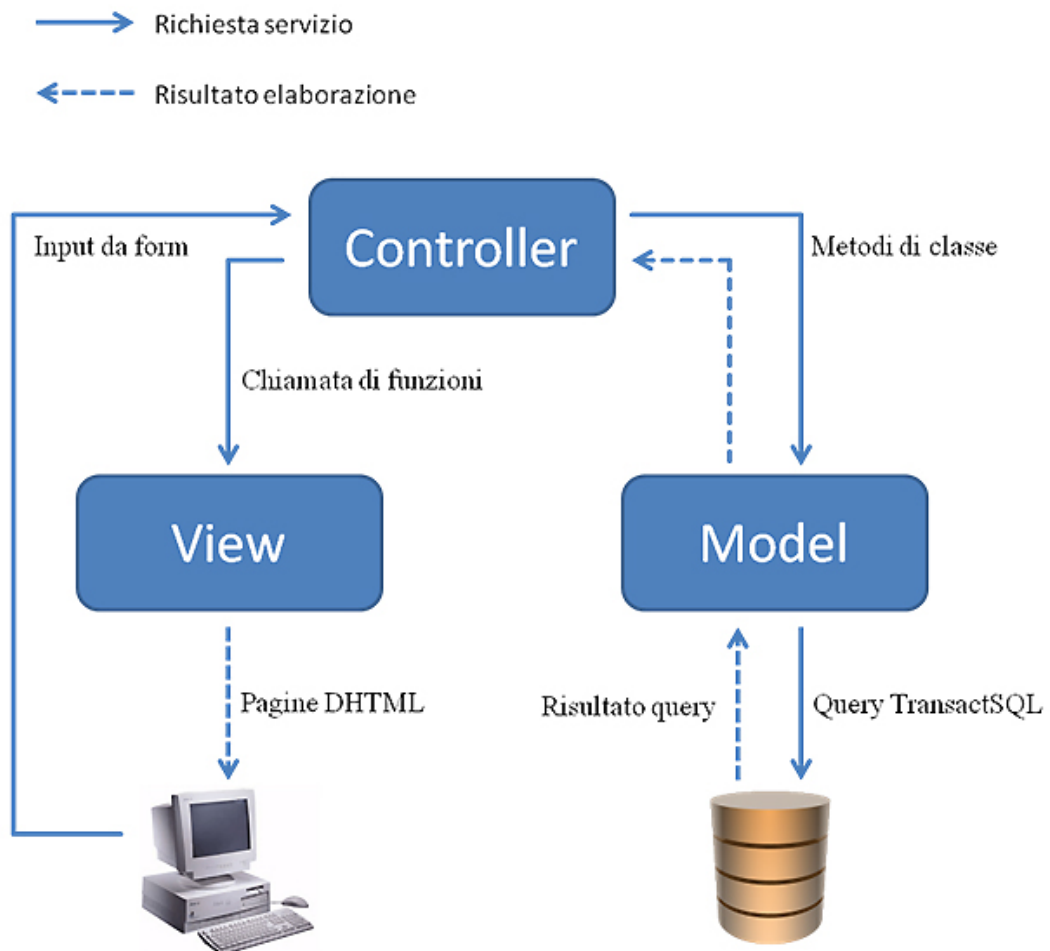


Figura 57: Architettura di SyndeV

5.4 Database Syndev

SQL Server Microsoft è il DBMS scelto per gestire il database di Syndev quindi le query per la creazione delle tabelle, i trigger e le interrogazioni per l'accesso ai dati sono state scritte in *Transact SQL*, estensione proprietaria del linguaggio SQL sviluppata da *Microsoft*.

Lo script *Transact SQL* per la creazione delle tabelle elencate in *4.5.1 Traduzione in schema relazionale* è in *Appendice B*, si può notare che compaiono dei *check constraint* corrispondenti agli stereotipi *enum* delle classi illustrate in *4.3 Class diagram*. Sempre in *Appendice B* viene mostrato il codice per la creazione dei trigger elencati in *4.5.2 Trigger*.

PHP fornisce delle funzioni per l'accesso diretto a *SQL Server*. Per poter funzionare su piattaforma *Win32* è necessario che l'intero pacchetto *MS SQL Client Tools* fornito da *Microsoft* sia installato sulla macchina server, mentre su piattaforma *Unix/Linux* occorre compilare e installare la libreria *FreeTDS*^[42]. Tra le funzioni di uso più comune di PHP vi sono:

- `mssql_connect(server, [utente], [password])`
ha come input l'URL del server e le informazioni di connessione, tenta di connettersi e, se va a buon fine, ritorna l'identificatore della connessione. È fornita anche una versione per aprire connessioni persistenti (`mssql_pconnect`);
- `mssql_select_db(database[connessione])`
seleziona il database MS SQL passato come parametro;
- `mssql_query(query[connessione])`
acquiesce ed esegue la query SQL passata come parametro e restituisce l'identificativo del risultato;
- `mssql_fetch_array(id_risultato, tipo_array)`
restituisce un'array associativo, cioè con gli elementi identificati dai nomi delle colonne, numerico o entrambi corrispondente alla prima riga del risultato passato come parametro e alla successiva riga nella chiamata seguente;

- `mssql_close([connessione])`
chiude l'ultima connessione aperta o la connessione passata come parametro.

Un'alternativa all'utilizzo delle funzioni per *Microsoft SQL Server*, da me effettivamente utilizzata, sono le funzioni di accesso a fonti ODBC che sono installate e abilitate di default in *PHP 5.2.5*^[45]. *ODBC* è l'acronimo di *Open Database Connectivity*, fornisce delle API standard per l'accesso alle basi di dati tramite un driver specifico per ogni fonte, in questo modo si può accedere a DBMS eterogenei utilizzando lo stesso codice. Quindi è stato necessario installare il driver per accesso a *Microsoft SQL Server (Microsoft SQL Server Native Client)* e aggiungere la fonte *ODBC* associata al server. Le funzioni utilizzate nel programma sono analoghe a quelle descritte sopra:

- `odbc_connect` e `odbc_pconnect` per la connessione alla fonte ODBC;
- `odbc_exec` per l'esecuzione delle interrogazioni;
- `odbc_fetch_array` per il reperimento dei risultati di `odbc_exec`;
- `odbc_close` per la chiusura della connessione.

ODBC è stato implementato in più versioni per diverse piattaforme, si può citare ad esempio *Microsoft ODBC* per *Windows* e *iODBC* per molte altre piattaforme^[43]. Quindi, allo stato attuale di sviluppo, *SyndeV* può risiedere su un server web con interprete *PHP* di un qualsiasi sistema operativo che fornisca le API *ODBC* e abbia il driver adatto installato.

6. Conclusioni e sviluppi futuri

Il progetto ha avuto come scopo la realizzazione di un'applicazione gestionale per l'azienda ospitante e gran parte degli obiettivi dello stage sono stati raggiunti. Per lo sviluppo dell'applicazione è stato necessario procedere secondo un modello di sviluppo che ha previsto le fasi di:

- **analisi dei requisiti**

si è effettuata l'analisi dei programmi utilizzati in azienda per la gestione e il manuale di qualità e si è discusso con il personale e gli amministratori per determinare chiaramente quello che l'applicazione andrà a fare;

- **progetto**

si è definito come il programma farà quello che è stato stabilito nell'analisi descrivendo il risultato della fase di progettazione utilizzando il modello *UML* per illustrare le entità che faranno parte del sistema e come dovranno interagire; il modello *Entity-Relationship* e quello relazionale sono stati invece utilizzati per descrivere la base di dati;

- **implementazione**

a partire dal progetto prodotto nella fase precedente si è realizzata l'applicazione utilizzando il linguaggio *PHP*, le tecnologie *DHTML* e interfacciandosi ad un DBMS *Microsoft SQL Server* per la gestione della base di dati.

Le prime due fasi del modello sono state completate, i requisiti sono stati definiti (v. capitoli 2 e 3) e il progetto è stato completato (v. capitolo 4).

La fase di implementazione non è stata invece terminata. Attualmente *SyndeV* fornisce le funzionalità per la gestione dei processi di sviluppo in *Sistema Qualità*:

1.g.i Commessa

1.g.ii Fasi e sottofasi

1.g.iii offerta

- 1.g.iv Fatturazione*
- 1.g.v Consolidato*
- 1.g.vi Stati Commessa-Offerta e interazioni*
- 1.g.vii Cartellino*
- 1.g.x Utenza*
- 1.g.xi Altri requisiti*

Resta quindi da sviluppare tutta la parte dell'applicazione relativa alla produzione di report per il controllo di gestione (v. *3.5Requisiti Syndev*), la creazione di commesse di integrazione (v. *2.6.7 Integrazione*) e la visualizzazione dei Gantt chart di supporto alla pianificazione (v. *2.6.9 Gantt chart*).

Poiché il programma è stato progettato per l'uso in intranet le problematiche relative alle prestazioni dell'applicazione in termini di velocità e di sicurezza non sono state affrontate. Nel momento in cui si voglia rendere l'applicazione web accessibile dell'esterno dell'intranet aziendale un ulteriore sviluppo del progetto potrebbe consistere nell'utilizzo di *AJAX* per il miglioramento della velocità e delle funzioni di crittografia fornite da *PHP* (es. md5) per la sicurezza.

Si necessita inoltre di una fase di test.

Riepilogo Attività in Sede														Controllo				
range anno: 2020 data gg: 01/01/2021																		
FILTRI: tutti tutti tutti tutti RP																		
		offerta		attività		fatturato		delta		percentuali delta		semaforo delta ricavo	percentuali per livello					
		gg off	mp off	gg eff	costo eff	gg fat	mp fat	gg fat-gg eff	fat-cost	%gg	%ricavo		RP/CL	RP	AIB	ATTIV	CLIENTE	
CED	P	H	AG	400	150.000,00	320	96.000,00	300	110.000,00	-20	-1.000,00	-5,67%	-0,91%		66%			
CRP	P	H	AG	400	150.000,00	320	96.000,00	300	110.000,00	-20	14.000,00	-5,67%	12,73%		34%			
TOT RP			AG	300.000,00	640,00	192.000,00	600,00	220.000,00	-40,00	13.000,00	-5,67%	5,91%		66%				
CRP	P	H	SC	200	100.000,00	80	70.000,00	100	90.000,00	20	20.000,00	20,00%	22,22%		34%			
TOT RP			SC	100.000,00	80	70.000,00	100	90.000,00	20	20.000,00	20,00%	22,22%		12,00%				
CED	P	W	CF	400	150.000,00	320	96.000,00	300	110.000,00	-20	-1.000,00	-5,67%	-0,91%		66%			
CRP	G	H	CF	400	150.000,00	320	96.000,00	300	110.000,00	-20	14.000,00	-5,67%	12,73%		34%			
TOT RP			CF	800	300.000,00	640,00	192.000,00	600,00	220.000,00	-40,00	13.000,00	-5,67%	5,91%		66%			
TOT AMBIENTE			H	1000	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	10.000,00	10,00%	4,00%		15,00%			
TOT AMBIENTE			W												12,00%			
TOT ATTIVITA			P														75,00%	
TOT ATTIVITA			G														25,00%	
TOT CLIENTE			CRP	1200	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	50.000,00	10,00%	20,00%					86,00%
TOT CLIENTE			CED	1200	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	50.000,00	10,00%	20,00%					14,00%
TOTALE COMPLESSIVO SOLO PRODUZIONE DIRETTA																		
TOTALE COSTI INDIRETTI						12	21.000,00											
TOTALE COMPLESSIVO COSTI DIRETTI + INDIRETTI																		

Figura 60: Appendice A, controllo - riepilogo commesse interne

Riepilogo Attività Esterna														Controllo				
range anno: 2020 data gg: 01/01/2021																		
FILTRI: tutti tutti tutti tutti																		
		offerta		attività		fatturato		delta		percentuali delta		semaforo delta ricavo	percentuali					
		gg off	mp off	gg eff	costo eff	gg fat	mp fat	gg fat-gg eff	fat-cost	%gg	%ricavo		proced	CLIENTE	AIB	ATTIVITA		
CED	G	H	te	400	150.000,00	320	96.000,00	300	110.000,00	-20	-1.000,00	-5,67%	-0,91%		66%			
CED	G	H	cc	400	150.000,00	320	96.000,00	300	110.000,00	-20	14.000,00	-5,67%	12,73%		34%			
TOT CLIENTE			CED	300.000,00	640,00	192.000,00	600,00	220.000,00	-40,00	13.000,00	-5,67%	5,91%		66%				
CRP	G	H	SC	200	100.000,00	80	70.000,00	100	90.000,00	20	20.000,00	20,00%	22,22%		34%			
TOT CLIENTE			CRP	100.000,00	80	70.000,00	100	90.000,00	20	20.000,00	20,00%	22,22%		12,00%				
TOT AMBIENTE			H	1000	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	10.000,00	10,00%	4,00%		15,00%			
TOT AMBIENTE			W												12,00%			
TOT ATTIVITA			P														75,00%	
TOT ATTIVITA			G														25,00%	
TOTALE COMPLESSIVO SOLO PRODUZIONE DIRETTA																		
TOTALE COSTI INDIRETTI						12	21.000,00											
TOTALE COMPLESSIVO COSTI DIRETTI + INDIRETTI																		

Figura 61: Appendice A, controllo - riepilogo commesse esterne

Lista Varie										
range	anno	dal	al	data gg						
	01/01/2007									
FILTRI	AREA									
	tutti/I/E									
codice		descrizione				percentuali				
		risorsa	ore	gg	costo	RIS				
		gg eff	costo eff	gg fatt		VAR	AREA	D/I		
COSTI VARIE ATTIVITA' DIRETTE										
I	XSYNVAR	Varie non imputabili								
		Rossi Costanza	54	6,75	1.200,10	66%				
		Arianna Larocca	12	1,50	344,44	34%				
	TOTALE XSYNVAR		66,00	8,25	1.544,44			88,00%		
I	XSYNFOR	Formazione								
		Barbara Pesenti	4	0,50	120,00	100%				
	TOTALE XSYNFOR		4,00	0,50	120,00			12,00%		
TOTALE AREA I			70,00	8,75	1.800,44			60,00%		
E	XSYNRIU	Formazione								
		Sabrina Sbalbi	4	0,50	120,00	100%				
	TOTALE XSYNRIU		4,00	0,50	120,00			100,00%		
TOTALE AREA E			70,00	8,75	1.800,44			40,00%		
TOTALE COSTI VARIE DIRETTE			100,00	12,00	3.300,00				20,00%	
COSTI VARIE ATTIVITA' INDIRETTE										
V	XSYNQUA	Qualità								
		Patrizia Egano	12	1,50	344,44	100%				
	TOTALE XSYNQUA		12,00	0,05	344,44			45,00%		
V	XSYNNAC	Non Allocate								
		Adriano Gotri	12	1,50	344,44	100%				
		Rossi Costanza	4	0,50	120,00	100%				
	TOTALE XSYNNAC		12,00	0,05	344,44			55,00%		
TOTALE COSTI VARIE INDIRETTE									80,00%	
TOTALE										
TOTALE COSTI VARIE										

Figura 62: Appendice A, controllo - riepilogo commesse indirette

Riepilogo Attività Comprensiva														Controllo				
anno: <input type="text"/> da: <input type="text"/> al: <input type="text"/> cliente: <input type="text"/> attività: <input type="text"/> ambiente: <input type="text"/> gg off: <input type="text"/>														percentuali				
AREA	cliente	attività	ambiente	gg off	mg off	gg on	costo off	gg let	mg let	gg let gg off	tot-cost	%gg	%costo	delta ricavo	ambiti	classe	sub tot	AREA
INTERNE																		
I	CED	P	H	400	150.000,00	320	96.000,00	300	110.000,00	-20	-1.000,00	-0,97%	-0,91%	X	66%			
I	CED	P	W	400	150.000,00	320	96.000,00	300	110.000,00	-20	-1.000,00	-0,97%	-0,91%	X	44%			
TOT CLIENTE CED				800	300.000,00	640,00	192.000,00	600,00	220.000,00	-40,00	-2.000,00	-0,97%	-0,91%	X	66%			
I	CRP	P	H	400	150.000,00	320	96.000,00	300	110.000,00	-20	-1.000,00	-0,97%	-0,91%	X	33%			
I	CRP	P	W	400	150.000,00	320	96.000,00	300	110.000,00	-20	-1.000,00	-0,97%	-0,91%	X	33%			
I	CRP	G	H	400	150.000,00	320	96.000,00	300	110.000,00	-20	14.000,00	-0,97%	12,73%	X	34%			
TOT CLIENTE CRP				800	300.000,00	640,00	192.000,00	600,00	220.000,00	-40,00	13.000,00	-0,97%	5,91%	X	66%			
TOT AMBIENTE H				1000	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	10.000,00	10,00%	4,00%	X	85,00%			
TOT AMBIENTE W				10	4.500,00	20,00	20.000,00								15,00%			
TOT ATTIVITA P				1000	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	10.000,00	10,00%	4,00%	X	15,00%			
TOT ATTIVITA G				10	4.500,00	20,00	20.000,00								15,00%			
TOT INTERNE				1000	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	10.000,00	10,00%	4,00%	X	85,00%			
Tot. Varie Interne				10	4.500,00	20,00	20.000,00								15,00%			
TOT AREA INTERNE																		35,00%
ESTERNE																		
E	CED	G	H	400	150.000,00	320	96.000,00	300	110.000,00	-20	-1.000,00	-0,97%	-0,91%	X	66%			
E	CED	G	W	40	15.000,00	32	9.600,00	30	11.000,00	-2	-100,00	-0,97%	-0,91%	X	66%			
TOT CLIENTE CED				800	165.000,00	352,00	105.600,00	330,00	121.000,00	-22,00	-1.100,00	-0,97%	-0,91%	X	66%			
TOT AMBIENTE H				1000	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	10.000,00	10,00%	4,00%	X	15,00%			
TOT AMBIENTE W				10	4.500,00	20,00	20.000,00								15,00%			
TOT ATTIVITA P				1000	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	10.000,00	10,00%	4,00%	X	15,00%			
TOT ATTIVITA G				10	4.500,00	20,00	20.000,00								15,00%			
TOT ESTERNE				1000	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	10.000,00	10,00%	4,00%	X	15,00%			
Tot. Varie Esterne				10	4.500,00	20,00	20.000,00								15,00%			
TOT AREA ESTERNE																		60,00%
TOT Varie Indirette																		5,00%
TOTALE COMPLESSIVO																		
PER CLIENTE																		
TOT CLIENTE CRP				1200	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	50.000,00	10,00%	20,00%	X	86,00%			
TOT CLIENTE CED				1200	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	50.000,00	10,00%	20,00%	X	34,00%			
PER ATTIVITA'																		
TOT AMBIENTE P				1200	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	50.000,00	10,00%	20,00%	X	86,00%			
TOT AMBIENTE G				1200	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	50.000,00	10,00%	20,00%	X	34,00%			
PER AMBIENTE																		
TOT AMBIENTE H				1200	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	50.000,00	10,00%	20,00%	X	86,00%			
TOT AMBIENTE W				1200	450.000,00	360,00	200.000,00	400,00	250.000,00	40,00	50.000,00	10,00%	20,00%	X	34,00%			

Figura 63: Appendice A, controllo - riepilogo totale

Lista Risorse														Controllo riepilogo risorse							
RANGE: anno: <input type="text"/> da: <input type="text"/> al: <input type="text"/> FILTRI: <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>														percentuali su totale							
risorsa	ore	dirette	costo	ore	indirette	costo	totale lavorato	assente	realizzazione	gg let	mg let	%gg	DELTA	% cost	% gg	% fatturato	proble	costo	gg	fatturate	importo
Agente Patrica	1.200,0	33.000,0	17,0	100,0	152,15	33.100,0	4,0	100,0	37.000,0	7,86	1.900,00	4,9%	X	5,1%	X	34%	34%	34%	34%		
Asher Daniele	1.200,0	22.000,0	0,0	0,0	150,00	22.000,0	0,0	140,0	20.000,0	-3,26	-2.000,00	-2,3%	X	-10,0%	X	34%	34%	34%	34%		
TOTALE	2.400,00	5.500,00	17,00	100,00	302,00	55.100,00	12,00	300,00	57.000,00	-2,9	1.900,00	-0,7%	X	5,7%	X						

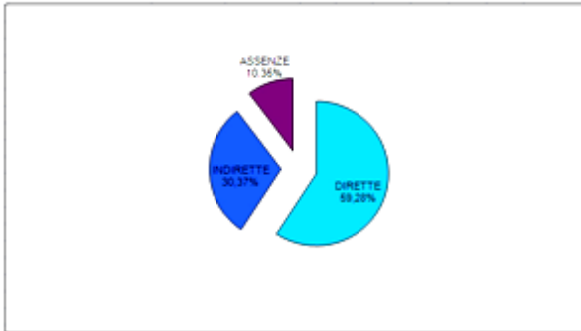
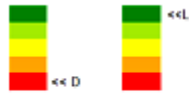
Figura 64: Appendice A, controllo - riepilogo risorse

Ore Risorsa		Controllo	
RANGE	anno	dal	al
		01/01/2007	data gg

Bertani Francesco

RIEPILOGO

tipologia	ore	%
PRODUZIONE	1.188,5	59,28%
VARIE	609,0	
ASSENZE	207,5	
TOTALE LAVORATIVE	1.797,5	89,65%
TOTALE ORE	2.005,0	



DETTAGLIO

Presenze

	svolte		realizzazione		delta		% ore	% fatturato	% sul tot
	ore	costo	ore fatt	imp fatt	#ore	fab-cost			
CRP7S22	150,0	33.000,0	160,0	35.000,00	10,0	2.000,00	0,8%	X	77,0%
CRP7S28	15,0	3.000,0	16,0	3.500,00	1,0	200,00	0,6%	X	22,0%
XSYNVAR	1,0	300,0							1,0%
TOTALE	171,0	36.300,0	176,0	38.500,00	4,5	1.950,00	0,9%	X	100,0%

Assenze

	ore	% sul tot
YSYNFER	15,0	77,0%
YSYNDON	1,0	22,0%
YSYNMAL	4,0	1,0%
TOTALE	20,0	

Figura 65: Appendice A, controllo - dettaglio risorsa

Appendice B: codice T-SQL per la creazione delle tabelle e trigger

Codice per la creazione delle tabelle dello schema illustrato nel paragrafo 4.5.1
Tradizione in schema relazionale.

```
create table cliente
(
    codiceCliente varchar(15) primary key,
    descrizione varchar(100));

create table attivita
(
    codiceAttivita varchar(15) primary key,
    descrizione varchar(100));

create table ambiente
(
    codiceAmbiente varchar(15) primary key,
    descrizione varchar(100));

create table competenza
(
    anno datetime primary key);

create table livello_rischio
(
    ordine tinyint primary key,
    livello varchar(100),
    autorizzazione char(25)
    constraint check_autorizzazione check
        (autorizzazione in
            ('amministrazione','responsabile sviluppo',
            'responsabile progetto')));

create table figura_professionale
(
    codiceFigura varchar(15) primary key,
    descrizione varchar(100));

create table procedura_progetto
(
    codiceProceduraProgetto varchar(15) primary key,
    descrizione varchar(100));

create table settore_attivita
(
    idSettoreAttivita varchar(15) primary key,
    nome varchar(100),
    nomePiano varchar(100),
    ordine smallint,
    ioStandard varchar(100),
    codDocPiano varchar(100));

create table tipo_fase
(
    idTipo varchar(15) primary key,
    idSettoreAttivita varchar(15)
    constraint fk_settore_attivita
        foreign key (idSettoreAttivita) references settore_attivita,
    ordine smallint,
    descrizione varchar(100));

create table risorsa
(
    codiceRisorsa varchar(15) primary key,
    nome varchar(25),
    cognome varchar(25),
    amministratore char(1)
    constraint check_amministratore check (amministratore in ('s','n')),
    username varchar(25) unique,
    password varchar(25));
```

```

create table responsabile
(
    codiceResponsabile varchar(15) primary key,
    codiceRisorsa varchar(15),
    descrizione varchar(100),
    progettoSviluppo char(1),
    constraint fk_responsabile_risorsa
        foreign key (codiceRisorsa) references risorsa,
    constraint check_progetto_sviluppo check (progettoSviluppo in
                                                ('p','s')));

create table costo_orario
(
    codiceRisorsa varchar(15),
    dataInizio datetime,
    dataFine datetime,
    internoEsterno char(1),
    costo float
    constraint pk_costo_orario
        primary key (codiceRisorsa,dataInizio,dataFine,internoEsterno),
    constraint fk_costo_orario_risorsa
        foreign key (codiceRisorsa) references risorsa,
    constraint check_interno_esterno check (internoEsterno in ('e','i')));

create table commessa
(
    codiceCommessa varchar(15) primary key,
    titolo varchar(100),
    stimaRealizzazione tinyint,
    dataStima datetime,
    area char(10),
    dataInizio datetime,
    dataFine datetime,
    note varchar(1000),
    prerequisiti varchar(1000),
    tipologia char(25),
    sviluppoAssistenza char(10),
    statoSviluppo char(10),
    statoCommessa char(20),
    codiceResponsabile varchar(15),
    codiceCliente varchar(15),
    livelloRischio tinyint,
    codiceAttivita varchar(15),
    codiceAmbiente varchar(15),
    codiceProceduraProgetto varchar(15),
    annoCompetenza datetime
    constraint fk_responsabile
        foreign key (codiceResponsabile) references responsabile,
    constraint fk_cliente foreign key (codiceCliente) references cliente,
    constraint fk_livello_rischio
        foreign key (livelloRischio) references livello_rischio,
    constraint fk_attivita foreign key (codiceAttivita) references attivita,
    constraint fk_ambiente foreign key (codiceAmbiente) references ambiente,
    constraint fk_procedura_progetto
        foreign key (codiceProceduraProgetto) references
            procedura_progetto,
    constraint fk_competenza foreign key (annoCompetenza) references
            competenza,
    constraint check_area check (area in ('interna','esterna','varia')),
    constraint check_sviluppo_assistenza
        check (sviluppoAssistenza in ('sviluppo','assistenza')),
    constraint check_tipologia
        check (tipologia in ('diretta di produzione','indiretta di
            produzione','indiretta','assenza')),
    constraint check_commissa_stato_sviluppo
        check (statoSviluppo in ('attiva','terminata')),

```

```

constraint check_stato_commissa
    check (statoCommissa in ('inserita','modificare','offerta
        emessa','offerta accettata',
        'offerta rifiutata',
        'in fatturazione',
        'fatturazione chiusa','chiusa')));

create table fase
(
    codiceCommissa varchar(15),
    codiceFase varchar(15),
    nome varchar(25),
    note varchar(1000),
    tipoFase varchar(15)
    constraint pk_fase primary key (codiceCommissa, codiceFase),
    constraint fk_fase_commissa foreign key (codiceCommissa) references
        commissa,
    constraint fk_tipo_fase foreign key (tipoFase) references tipo_fase);

create table sottofase
(
    codiceCommissa varchar(15),
    codiceFase varchar(15),
    codiceSottofase varchar(15),
    codiceRisorsa varchar(15),
    codiceFiguraProfessionale varchar(15),
    giorniStima float,
    inizio datetime,
    fine datetime,
    terminazione char(1),
    note varchar(1000)
    constraint pk_sottofase
        primary key (codiceCommissa, codiceFase,
            codiceSottofase, codiceRisorsa),
    constraint fk_sottofase_fase
        foreign key (codiceCommissa, codiceFase) references fase,
    constraint fk_sottofase_risorsa
        foreign key (codiceRisorsa) references risorsa,
    constraint fk_figura_professionale
        foreign key (codiceFiguraProfessionale) references
            figura_professionale,
    constraint check_terminazione check (terminazione in ('s','n')));

create table fattura
(
    codiceCommissa varchar(15),
    numero tinyint,
    effettivaPrevista char(1),
    data datetime,
    importo float,
    note varchar(1000)
    constraint pk_fattura primary key
        (codiceCommissa, numero, effettivaPrevista),
    constraint fk_fattura_commissa
        foreign key (codiceCommissa) references commissa,
    constraint check_effettiva_prevista check (effettivaPrevista
        in('e','p')));

create table offerta
(
    codiceCommissa varchar(15) primary key,
    codiceAccettazione varchar(15),
    data datetime,
    importo float,
    dataAccettazione datetime,
    giorniAccettati float,
    importoAccettato float,
    prerequisiti varchar(1000),

```

```

descrizione varchar(1000),
dettaglioAttivita varchar(1000),
interfacce varchar(1000),
inviareA varchar(1000),
statoOfferta varchar(15),
note varchar(1000)
constraint fk_offerta_commissa
    foreign key (codiceCommissa) references commessa,
constraint check_stato_offerta
    check (statoOfferta in
        ('precompilata','emessa','accettata','rifiutata','modificare')));

create table giorni_offerta
(
    codiceCommissa varchar(15),
    codiceFigura varchar(15),
    numeroGiorni float
constraint pk_giorni_offerta primary key (codiceCommissa,codiceFigura),
constraint fk_giorni_offerta_commissa
    foreign key (codiceCommissa) references commessa,
constraint fk_giorni_offerta_figura_professionale
    foreign key (codiceFigura) references figura_professionale);

create table consolidato
(
    codiceCommissa varchar(15) primary key,
    titolo varchar(100),
    statoSviluppo char(10),
    livelloRischio tinyint,
    dataInizio datetime,
    dataFine datetime,
    codiceArea char(10),
    codiceCliente varchar(15),
    codiceAttivita varchar(15),
    codiceAmbiente varchar(15),
    responsabileProgetto varchar(15),
    codiceProceduraProgetto varchar(15),
    note varchar(1000),
    prerequisiti varchar(1000),
    descrizioneOfferta varchar(1000),
    dettaglioAttivita varchar (1000),
    interfacce varchar(1000),
    inviareA varchar(1000)
constraint fk_consolidato_commissa
    foreign key (codiceCommissa) references commessa);

create table fase_consolidata
(
    codiceConsolidato varchar(15),
    codiceFaseConsolidata varchar(15),
    nome varchar(25),
    note varchar(1000),
    tipoFase varchar(15)
constraint pk_fase_consolidata
    primary key (codiceConsolidato,codiceFaseConsolidata),
constraint fk_consolidato
    foreign key (codiceConsolidato) references consolidato);

create table sottofase_consolidata
(
    codiceConsolidato varchar(15),
    codiceFaseConsolidata varchar(15),
    codiceSottofaseConsolidata varchar(15),
    codiceRisorsa varchar(15),
    codiceFiguraProfessionale varchar(15),
    giorniStima float,
    inizio datetime,

```



```

fine datetime,
terminazione char(1),
note varchar(1000)
constraint pk_sottofase_consolidata
    primary key (codiceConsolidato,codiceFaseConsolidata,
                codiceSottofaseConsolidata),
constraint fk_fase_consolidata
    foreign key (codiceConsolidato,codiceFaseConsolidata)
                references fase_consolidata);

create table cartellino
(
    codiceCommessa varchar(15),
    codiceFase varchar(15),
    codiceSottofase varchar(15),
    codiceRisorsa varchar(15),
    data datetime,
    ore float
constraint pk_cartellino primary key (codiceCommessa,codiceFase,data),
constraint fk_sottofase_risorsa_cartellino
    foreign key
        (codiceCommessa,codiceFase,codiceSottofase,codiceRisorsa)
        references sottofase);

```

Il seguente codice si riferisce al trigger 1 di 4.5.2 *Trigger*:

```

create trigger tr_creazione_commessa
on commessa
after insert
as
begin
    declare @codiceCommessa varchar(15);

    select @codiceCommessa =
    (
        select codiceCommessa
        from inserted);

    insert offerta
    values
        (@codiceCommessa, '', '', 0, '', 0, 0, '',
        '', '', '', 'precompilata', '');
end;

```

Il seguente codice si riferisce al trigger 2 di 4.5.2 *Trigger*:

```

create trigger tr_emissione_offerta
on offerta
after update
as
begin
    declare @statoOfferta varchar(15);
    declare @codiceCommessa varchar(15);

    select @statoOfferta =
    (
        select statoOfferta
        from inserted);

```

```

if @statoOfferta = 'emessa'
begin
    select @codiceCommessa =
    (
        select codiceCommessa
        from inserted)

    update commessa
    set statoCommessa = 'offerta emessa'
    where codiceCommessa = @codiceCommessa

    insert consolidato
    select
        c.codiceCommessa,
        c.titolo,
        c.statoSviluppo,
        c.livelloRischio,
        c.dataInizio,
        c.dataFine,
        c.area,
        c.codiceCliente,
        c.codiceAttivita,
        c.codiceAmbiente,
        c.codiceResponsabile,
        c.codiceProceduraProgetto,
        c.note, c.prerequisiti,
        o.descrizione,
        o.dettaglioAttivita,
        o.interfacce,
        o.inviareA
    from commessa c, offerta o
    where c.codiceCommessa = o.codiceCommessa
    and c.codiceCommessa = @codiceCommessa;

    insert fase_consolidata
    select *
    from fase
    where codiceCommessa = @codiceCommessa;

    insert sottofase_consolidata
    select * from sottofase
    where codiceCommessa = @codiceCommessa;
end;
end;

```

Il seguente codice si riferisce al trigger 3 di 4.5.2 *Trigger*:

```

create trigger tr_accettazione_offerta
on offerta
after update
as
begin
    declare @statoOfferta varchar(15);
    declare @codiceCommessa varchar(15);

    select @statoOfferta =
    (
        select statoOfferta
        from inserted);

    if @statoOfferta = 'accettata'
    begin
        select @codiceCommessa =
        (
            select codiceCommessa
            from inserted);
    end;
end;

```

```

        update commessa
        set statoCommessa = 'offerta accettata'
        where codiceCommessa = @codiceCommessa;
    end;
end;

```

Il seguente codice si riferisce al trigger 4 di 4.5.2 *Trigger*:

```

create trigger tr_rifiuto_offerta
on offerta
after update
as
    begin
        declare @statoOfferta varchar(15);
        declare @codiceCommessa varchar(15);

        select @statoOfferta =
        (
            select statoOfferta
            from inserted);

        if @statoOfferta = 'rifiutata'
        begin
            select @codiceCommessa =
            (
                select codiceCommessa
                from inserted)

            update commessa
            set statoCommessa = 'offerta rifiutata' ,
            statoSviluppo = 'terminata'
            where codiceCommessa = @codiceCommessa
        end;
    end;
go

```

Il seguente codice si riferisce al trigger 5 di 4.5.2 *Trigger*:

```

create trigger tr_modifica_offerta
on offerta
after update
as
    begin
        declare @statoOfferta varchar(15);
        declare @codiceCommessa varchar(15);

        select @statoOfferta =
        (
            select statoOfferta
            from inserted);

        if @statoOfferta = 'modificare'
        begin
            select @codiceCommessa =
            (
                select codiceCommessa
                from inserted)

            update commessa
            set statoCommessa = 'modificare'
            where codiceCommessa = @codiceCommessa
        end;
    end;

```

```

update offerta
set statoOfferta = 'precompilata'
where codiceCommessa = @codiceCommessa

delete from sottofase_consolidata
where codiceConsolidato = @codiceCommessa;

delete from fase_consolidata
where codiceConsolidato = @codiceCommessa;

delete from consolidato
where codiceCommessa = @codiceCommessa;
end;
end;

```

Il seguente codice si riferisce ai trigger 6 e 7 di 4.5.2 *Trigger*:

```

create trigger tr_fatturazione
on fattura
after insert
as
begin
declare @numeroFattureEffettive int;
declare @codiceCommessa varchar(15);
declare @ammontareFatturePreviste float;
declare @ammontareFattureEffettive float;

select @codiceCommessa =
(
select codiceCommessa
from inserted)

select @numeroFattureEffettive =
(
select count(*)
from fattura
where codiceCommessa = @codiceCommessa
and effettivaPrevista = 'e')

if @numeroFattureEffettive <> 0
begin
select @ammontareFatturePreviste =
(
select sum(importo)
from fattura
where codiceCommessa = @codiceCommessa
and effettivaPrevista = 'p')

select @ammontareFattureEffettive =
(
select sum(importo)
from fattura
where codiceCommessa = @codiceCommessa
and effettivaPrevista = 'e')

if @ammontareFattureEffettive >=
@ammontareFatturePreviste
begin
update commessa
set statoCommessa = 'fatturazione
chiusa'
where codiceCommessa = @codiceCommessa
end;
end;

```

```

                                else
                                  begin
                                    update commessa
                                      set statoCommessa = 'in fatturazione'
                                      where codiceCOmmessa = @codiceCommessa
                                  end;
                                end;
end;
```

Riferimenti bibliografici

- [1] UNI CEI EN 45020, punto 1.1
- [2] ISO 8402:1995
- [3] ISO 9000:2000
- [4] La misurazione della qualità del software, Gianluigi Raiss; CNIPA
- [5] ISO/IEC CD 25012: Data quality model – Domenico Natale; CNIPA
- [6] ISO 9001:2000, 0.2
- [7] ISO 9001:2000, 4.1
- [8] ISO 9001:2000, 6
- [9] ISO 9001:2000, 4.2
- [10] ISO 9001:2000, 7
- [11] Manuale della qualità – Synago srl, 7.4
- [12] Manuale della qualità – Synago srl, 7.6
- [13] ISO 9001:2000, 7.1
- [14] ISO 9001:2000, 7.2
- [14] ISO 9001:2000, 7.3
- [15] ISO 9001:2000, 7.5
- [16] ISO 9001:2000, 5
- [17] ISO 9001:2000, 8
- [18] Manuale della qualità – Synago srl, 7.1
- [19] Manuale della qualità – Synago srl, 7.2
- [20] Manuale della qualità – Synago srl, 7.3
- [21] Manuale della qualità – Synago srl, 7.4
- [22] Manuale della qualità – Synago srl, 7.5
- [23] Manuale della qualità – Synago srl, 7.3.1
- [24] Manuale della qualità – Synago srl, 7.3.2
- [25] Manuale della qualità – Synago srl, 7.3.2.5
- [26] Manuale della qualità – Synago srl, 7.3.4
- [27] Manuale della qualità – Synago srl, 7.3.5
- [28] Analisi dei costi e contabilità industriale – Alessandro Tullio; IPSOA, 1.1
- [29] Corso di economia aziendale, Vol.1 – Paola Miolo Vitali; Giappichelli, III.5.1
- [30] Economia aziendale – Airoldi, Brunetti, Coda; Il Mulino, p. 434

- [31] Analisi dei costi e contabilità industriale – Alessandro Tullio; IPSOA, 1.4
- [32] Corso di economia aziendale, Vol.1 – Paola Miolo Vitali; Giappichelli, III.5.2
- [33] Corso di economia aziendale, Vol.1 – Paola Miolo Vitali; Giappichelli, III.5.3
- [34] Analisi dei costi e contabilità industriale – Alessandro Tullio; IPSOA, 4
- [35] Analisi dei costi e contabilità industriale – Alessandro Tullio; IPSOA, 9.2
- [36] Analisi dei costi e contabilità industriale – Alessandro Tullio; IPSOA, 9.4.2
- [37] Progetto e realizzazione di una intranet aziendale con applicazione web Ajax – Orazio Valerio, tesi di laurea pubblicata su *dbgroup.unimo.it*, 2.1
- [38] Progetto e realizzazione di una intranet aziendale con applicazione web Ajax – Orazio Valerio, tesi di laurea pubblicata su *dbgroup.unimo.it*, 2.2
- [39] w3schools.com/dhtml/
- [40] apachefriends.org/en/xampp.html
- [41] java.sun.com/blueprints/patterns/MVC-detailed.html
- [42] it.php.net/mssql
- [43] iodbc.org/
- [44] <http://it2.php.net/zend-engine-2.php>
- [45] <http://it2.php.net/odbc>
- [46] <http://www.microsoft.com/sql/default.msp>