

FABIO PALESE

Università degli studi di Padova

LAUREA TRIENNALE IN INGEGNERIA INFORMATICA

RELAZIONE DEL TIROCINIO

REALIZZAZIONE DI UNA SOLUZIONE
WEB A UN CASO PRATICO DEL
PROBLEMA DEL COMMESSE
VIAGGIATORE

RELATORE: PROF. MARCELLO DALPASSO

RELAZIONE DEL TIROCINIO

Luglio 2011

Fabio Palese: *Realizzazione di una soluzione web a un caso pratico del problema del commesso viaggiatore*, Relazione del tirocinio, © Luglio 2011.

E-MAIL:

fabio.palese@gmail.com

Questo documento presenta la relazione del tirocinio da me svolto presso Sphera Engineering and Communication SRL, con inizio in data 15/11/2010 e termine in data 19/01/2011, e durata prevista di 250 ore.

Obiettivo del tirocinio è stato la creazione di un sistema web per la gestione di un'azienda di disinfestazione e del suo programma di interventi, per la cui implementazione si è ricorso anche all'applicazione di conoscenze acquisite dal corso di Ricerca Operativa 1.

INDICE

1	INTRODUZIONE	1
1.1	L'azienda	1
1.2	Il sistema IES	1
1.3	Obiettivi	2
1.4	Campo applicativo	2
2	ANALISI DEL PROGETTO	3
2.1	Approccio al progetto	3
2.2	Panoramica	3
2.2.1	Gestione Utenti e Login	4
2.2.2	Gestione Anagrafiche	4
2.2.3	Gestione Offerte e Contratti	5
2.2.4	Programmazione degli Interventi	6
2.2.5	Gestione del Calendario	11
2.3	Componenti	11
2.4	Interfaccia con l'utente	11
3	STRUMENTI E TECNOLOGIE	13
3.1	Linguaggi lato server	13
3.1.1	PHP	13
3.1.2	Doctrine ORM e DQL	14
3.1.3	MySQL e PHPMyAdmin	15
3.1.4	AJAX	15
3.2	Linguaggi lato client	15
3.2.1	HTML	15
3.2.2	CSS	17
3.2.3	Javascript	17
3.2.4	jQuery	18
3.2.5	jQuery UI	18
4	REALIZZAZIONE	19
4.1	Progettazione della base di dati	19
4.1.1	Progettazione Concettuale	19
4.1.2	Progettazione Logica	25
4.1.3	Progettazione Fisica	26
4.1.4	Comunicazione con il modulo applicativo	27
4.2	Realizzazione del sistema di gestione degli Utenti e Login	28
4.3	Realizzazione del gestionale delle anagrafiche	29
4.4	Realizzazione del Gestionale delle Offerte e dei Contratti	31
4.5	Realizzazione della Schermata di Gestione del Calendario	36
4.6	Realizzazione della Programmazione degli Interventi	37
4.6.1	Schedulazione automatica degli interventi	38
4.6.2	Schedulazione manuale	52
5	INTERFACCIA GESTIONALE WEB	55
6	CONCLUSIONI	65
6.1	Sviluppi futuri	65
6.2	Considerazioni sull'esperienza	65
6.3	Ringraziamenti	66
	BIBLIOGRAFIA	67

1

INTRODUZIONE

Questo capitolo ha lo scopo di descrivere l'azienda e il contesto in cui nasce la soluzione web in questione, nonché il suo obiettivo e campo di applicazione.

INDICE

1.1	L'azienda	1
1.2	Il sistema IES	1
1.3	Obiettivi	2
1.4	Campo applicativo	2

1.1 L'AZIENDA

L'Azienda *Sphera Engineering and Communication s.r.l.* si occupa della fornitura di soluzioni software per la gestione dei flussi informativi nelle aziende. Opera in un settore Business To Business e si rivolge in generale ad aziende che producono e commercializzano prodotti tecnici in mercati multinazionali, fornendo inoltre servizi di redazione documentale per risolvere le problematiche relative alla normalizzazione delle fonti dei dati, organizzazione ad oggetti delle informazioni, declinazione nei vari media di pubblicazione e collegamento di tutti i soggetti sulla rete (uff. MKT, uff. tecnico, sistemi informativi, traduttori etc.). I suoi servizi includono database publishing, sistemi internet per la gestione del pre e post vendita, studi di mercato (posizionamento e penetrazione), progetti di comunicazione istituzionale e campagne marketing. In questo contesto nasce e opera il *sistema IES*.

<http://www.spherastudio.com>

1.2 IL SISTEMA IES

IES è un software costituito da un sistema di applicazioni che, slegate dal gestionale di base e alimentate su web, s'introducono a moduli nelle parti aziendali interessate (vendita, assistenza tecnica, marketing, amministrazione, ecc...), servendosi di archivi e file utilizzabili sempre dai successivi moduli. In pratica fornisce un modello di gestione documentale su cui poi si basano altri applicativi aziendali quali il configuratore d'offerta, la gestione preventivi e ordini, la gestione budget e geomarketing, la previsione vendite

www.spherastudio.com/site/software/ies



Figura 1: Il logo aziendale

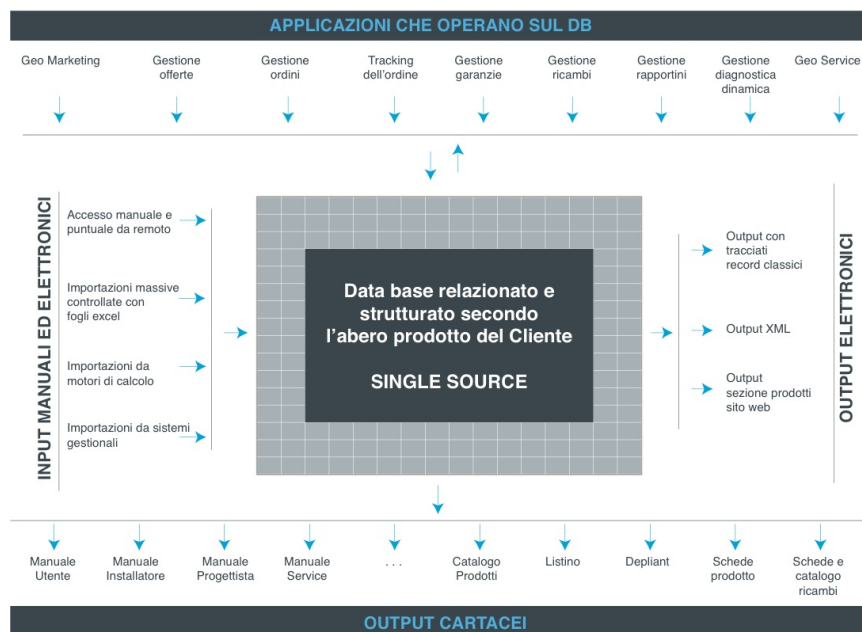


Figura 2: Schematizzazione della struttura di IES

e la gestione del post-vendita (interventi e ricambistica), mantenendo comunque una centralizzazione ed atomicità dei dati, quindi portando ad evitare problematiche dovute al loro disallineamento.

1.3 OBIETTIVI

L'obiettivo dell'esperienza è la realizzazione di un'applicazione web orientata alla gestione delle attività di un'azienda di disinfestazione, commissionata da una ditta esterna (d'ora in poi per comodità definita come committente). Le attività di tale azienda sono strutturate nel seguente modo: Si ha una prima fase di contatto con il cliente e sua conoscenza, successivamente viene avviata la programmazione di una serie di interventi tenendo conto di una serie di variabili come tipologia di parassita, ambiente, disponibilità del cliente, zona di intervento, ecc. Inoltre la programmazione degli interventi terrà conto di un insieme di mezzi, operatori e attrezzature disponibili per la messa in atto e vincoli a cui essi dovranno attenersi.

1.4 CAMPO APPLICATIVO

Il Modulo una volta ultimato andrà ad integrarsi con il sistema IES ed a complementarne le funzionalità nello specifico campo della gestione delle aziende di disinfestazione. Il suo utilizzo sarà direzionato su due funzionalità principali: una di gestione delle anagrafiche dati e quindi delle informazioni di base su cui andrà il sistema ad agire ed una seconda di gestione del calendario, dove sarà possibile regolare la schedulazione degli interventi su base mensile o giornaliera.

2

ANALISI DEL PROGETTO

Verrà qui effettuata una analisi dei requisiti del sistema da realizzare, che offrirà in particolare una panoramica delle funzionalità che il modulo applicativo dovrà essere in grado di offrire.

INDICE

2.1	Approccio al progetto	3
2.2	Panoramica	3
2.2.1	Gestione Utenti e Login	4
2.2.2	Gestione Anagrafiche	4
2.2.3	Gestione Offerte e Contratti	5
2.2.4	Programmazione degli Interventi	6
2.2.5	Gestione del Calendario	11
2.3	Componenti	11
2.4	Interfaccia con l'utente	11

2.1 APPROCCIO AL PROGETTO

Per il processo di sviluppo del software è stato scelto un modello incrementale; lo sviluppo avviene ciclicamente e per ogni funzionalità vengono seguite cinque fasi. Si ha una fase di *comunicazione* con il committente, in cui si definiscono i requisiti che la funzionalità dovrà avere; si procede con una fase di *pianificazione e modellazione*, in cui viene definita la struttura logica della base di dati ed avviene la stesura dello pseudocodice; una fase di *costruzione* in cui avviene l'effettiva programmazione della funzionalità e il suo successivo *collaudo* con correzione dei bug; infine si ha la fase di *dispiegamento*, in cui la funzionalità viene integrata con quelle preesistenti e il committente esprime un suo feedback a riguardo.

il modello incrementale

2.2 PANORAMICA

L'analisi della problematica comincia con un incontro con il committente, in cui viene effettuata una analisi dei requisiti: L'azienda committente si occupa di disinfestazione e della risoluzione di ogni tipo di problematica relativa a insetti, animali e parassiti presenti in qualsiasi tipo di ambiente. La richiesta iniziale è la seguente: ottenere una applicazione web per la gestione del calendario interventi e delle anagrafiche dati. In particolare l'analisi iniziale si sofferma su cinque punti principali:

- Login
- Anagrafiche
- Offerte e Contratti

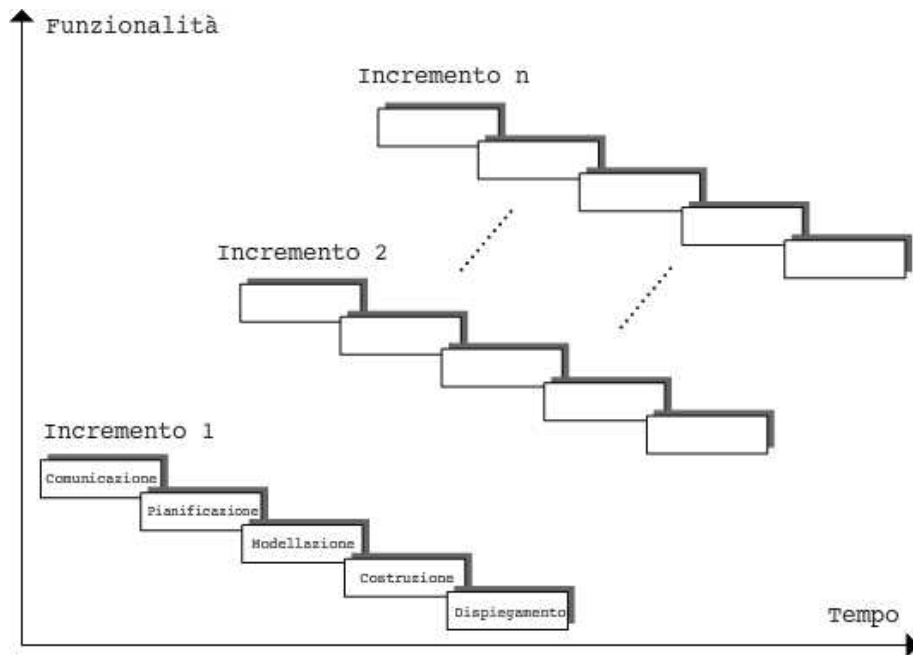


Figura 3: Schematizzazione del modello incrementale adottato per lo sviluppo

- Programmazione interventi
- Calendario

2.2.1 Gestione Utenti e Login

L'accesso all'applicazione dovrà essere ristretto solo ad utenti in possesso di un account, creato dall'amministratore del sistema, che provvede inoltre a regolare i permessi ed eventualmente a disabilitare gli account obsoleti. E' richiesta la gestione di tre categorie di utenti: *amministratore*, *operatore* e *cliente*.

2.2.2 Gestione Anagrafiche

E' richiesta una gestione delle anagrafiche per le seguenti entità:

CONTATTI Singoli o aziende che trattano (o hanno trattato) per entrare in relazioni contrattuali con il committente. Ognuno di essi può chiedere la disinfestazione di più ambienti o da più di un infestante.

CLIENTI Singoli o aziende che sono entrati in relazioni contrattuali con il committente. Ognuno di essi può chiedere la disinfestazione di più ambienti o da più di un infestante.

OPERATORI Le persone con cui l'azienda può effettuare gli interventi

INFESTANTI Comprende gli animali sui quali è possibile intervenire.

MEZZI I Mezzi utilizzabili dagli Operatori per gli spostamenti.

ATTREZZATURE L'Attrezzatura usata dagli operatori per effettuare gli interventi.

PRODOTTI I Prodotti utilizzabili nelle operazioni di disinfestazione

VINCOLI Sono delle condizioni che limitano l'intervento degli operatori nell'effettuare la disinfestazione. Sono suddivisibili nelle seguenti sottocategorie:

GENERICI Condizioni generali che riguardano ad esempio la presenza di un cane, di una persona disabile, ecc.

TEMPORALI Condizioni che limitano i giorni e gli orari in cui sono effettuabili gli interventi

POST-INTERVENTO Condizioni derivanti da un intervento precedente in una serie di interventi; generalmente vincolano la presenza (o l'assenza) di uno o più operatori, mezzi o attrezzature durante un intervento.

DI DESTINAZIONE dipendono dal luogo in cui viene effettuato l'intervento, ad esempio il piano del palazzo, la presenza di un cancello, ecc.

ZONE Sono delle suddivisioni, solitamente spaziali, del territorio atte a raggruppare i luoghi di intervento. Per questioni finanziarie e di gestione la schedulazione degli interventi dovrà, cercare di svilupparsi preferibilmente raggruppando interventi in luoghi appartenenti alla stessa zona.

TIPOLOGIE DI INTERVENTO Sono le procedure effettivamente attuabili dall'azienda. Ciascuna va a operare su un Infestante e può richiedere delle attrezzature e prodotti specifici. Inoltre hanno determinati operatori abilitati alla loro effettuazione.

FORNITORI Enti che forniscono all'azienda il materiale a sua disposizione (mezzi, attrezzature, prodotti).

2.2.3 Gestione Offerte e Contratti

E' richiesta una struttura per la gestione dei rapporti contrattuali relativi alla disinfestazione. La meccanica di base prevede in primo luogo il contatto con l'azienda da parte di un cliente (o potenzialmente tale) per la richiesta di un servizio di disinfestazione. L'Operatore provvederà quindi all'inserzione dei dati relativi al caso in questione quali generalità del cliente, luoghi da disinfestare e tipologie di intervento da eseguire, ed eventualmente un preventivo. Questi dati vanno a comporre l'offerta di servizio che viene proposta al cliente. L'offerta può subire ulteriori modifiche e, se il cliente decidesse di aderire, diventerebbe quindi un contratto. Il contratto quindi diventa attivo e si procede alla programmazione degli interventi necessari al suo adempimento. Sono richieste due tipologie di contratti: i *contratti standard* (denominati successivamente anche come O/C) che richiedono tutta la fase di contrattazione descritta precedentemente e i contratti di tipo *Spot*, che non richiedono la fase di contrattazione e solitamente sono usati per effettuare interventi a breve termine. E' richiesta inoltre la gestione di eventuali annullamenti, cancellazioni e rinnovamento dei contratti.

E' possibile quindi notare che nella loro gestione i contratti presentano una variazione di stati di cui si dovrà tenere conto durante la progettazione del sistema; gli stati in questione sono i seguenti:

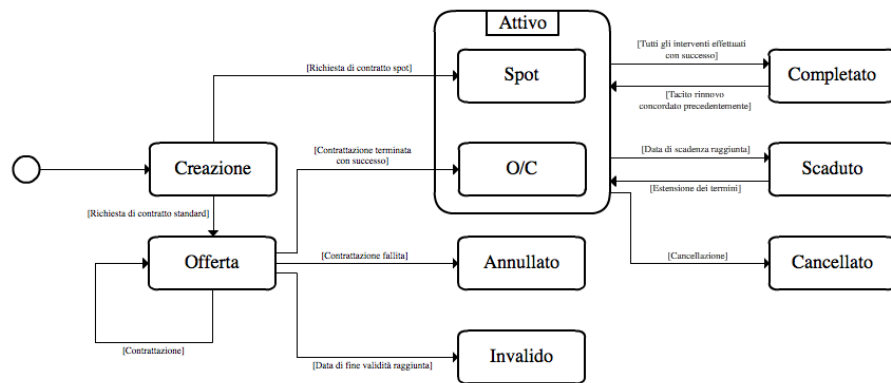


Figura 4: Diagramma di macchina a stati relativo al ciclo di vita di un offerta/contratto

- Una fase di *Creazione* dell'Offerta/Contratto;
- Uno stato di *Offerta*, in cui si stabilisce assieme al cliente/contatto;
- Uno stato di Contratto di tipo *Spot* in corso, che presenta l'effettiva attuazione delle operazioni di disinfestazione;
- Uno stato di Contratto di tipo *O/C* in corso, che anch'esso presenta l'effettiva attuazione delle operazioni di disinfestazione;
- Uno stato di *Annullamento* delle Offerte, nel caso in cui non sia stato possibile raggiungere un accordo con il cliente;
- Uno stato di *Invalidità*, in cui un Offerta ha raggiunto la fine del suo periodo di validità e non è più disponibile per essere portata a conclusione;
- Uno stato di *Completamento*, in cui un Contratto ha effettuato tutte le operazioni concordate e quindi può avere termine;
- Uno stato di *Scadenza*, in cui un Contratto non è riuscito a portare a termine tutte le operazioni concordate entro i termini specificati alla sua apertura;
- Uno stato di *Cancellazione*, in cui motivi particolari portano al termine della relazione contrattuale prima che essa riesca ad arrivare al termine.

Il ciclo di vita di un offerta/contratto è descrivibile con il diagramma di figura 4.

2.2.4 Programmazione degli Interventi

Un intervento viene effettuato dagli operatori dell'azienda con appropriati mezzi, effettuando una tipologia di intervento in un determinato luogo, rispettando eventuali vincoli definiti precedentemente su entità correlate e vincoli propri dell'intervento.

Sulla base dei dati inseriti precedentemente per i contratti dovrà essere effettuata la programmazione giornaliera degli interventi di disinfestazione

(da ora in poi definita come schedulazione); sono previste due modalità di esecuzione:

SCHEDULAZIONE AUTOMATICA Una modalità che mira ad effettuare la schedulazione giornaliera degli interventi, mirando ad ottenere un bilanciamento ottimale tra esigenze economiche dell'azienda (relative agli spostamenti degli operatori) ed efficacia della schedulazione, cercando quindi di minimizzare al tempo stesso i contratti che eccedono i termini di completamento. La schedulazione automatica deve fare ciò pur tenendo conto di tutti i vincoli che sono stati imposti durante le fasi di inserimento dati. Permette alla discrezione dell'utente di specificare termini e caratteristiche dei singoli interventi, di fatto lasciando la possibilità di personalizzare completamente il piano giornaliero degli interventi.

SCHEDULAZIONE MANUALE In questo contesto si vuole valutare i vincoli imposti nelle fasi precedenti come semplici avvisi: le modifiche che violano potenzialmente i vincoli riceveranno degli avvisi di allerta, ma saranno comunque permesse se l'utente, conscio di ciò, decidesse comunque di procedere.

E' comunque prevista la possibilità di alternare i due metodi precedenti secondo le proprie esigenze.

Il Problema del Commesso Viaggiatore applicato

E' opportuno effettuare una discussione più approfondita e teorica riguardante in particolare la prima modalità delle due, la schedulazione automatica; si adatterà un approccio per trattarla come istanza relativa alla problematica NP-difficile denominata come *problema del commesso viaggiatore*, ovvero un caso specifico di questo problema. Questo approccio deriva dalla forte similitudine tra i due problemi; l'operazione di schedulazione automatica infatti si porrà anche l'obiettivo di trovare i percorsi ottimali che collegano i luoghi nei quali gli operatori si dirigono ad effettuare gli interventi di disinfezione.

Al fine di comprendere meglio ciò in cui questo problema consiste e come si traduce nella pratica saranno discusse dettagliatamente le basi teoriche della ricerca operativa su cui esso si basa.

Il *problema del commesso viaggiatore* (o dall'inglese TSP, *Traveling Salesman Problem*) consiste nell'individuare un circuito hamiltoniano di costo minimo in un assegnato grafo orientato $G = (V, A)$.

Traveling Salesman Problem

Per *grafo orientato* si intende la coppia $G = (V, A)$, in cui V è un insieme finito di vertici e A è una famiglia di coppie orientate di vertici, dette *archi*. Ogni arco è una coppia del tipo (i, j) , con i e j vertici appartenenti all'insieme V . L'ordine in cui vengono definiti i due vertici è rilevante sui grafi orientati, poiché definisce la direzione dell'arco; si dice quindi che l'arco $(i, j) \in A$ esce dal vertice i ed entra nel vertice j , rispettivamente chiamati *coda* e *testa* dell'arco.

Grafi orientati

Per *circuito* si intende invece una sequenza (a_1, a_2, \dots, a_k) di archi consecutivi del tipo $a_1 = (v_1, v_2), a_2 = (v_2, v_3), \dots, a_k = (v_k, v_1)$, ovvero dove la coda di ogni arco coincida con la testa del successivo, con il ritorno nel vertice di partenza. In particolare il circuito si dice *hamiltoniano* se visita una e

Circuiti hamiltoniani

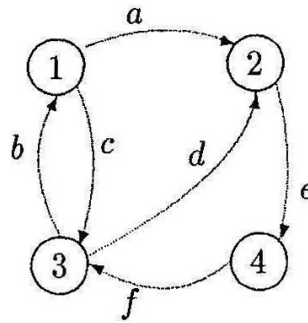


Figura 5: Esempio di grafo orientato; i vertici sono segnati con numeri, gli archi con lettere

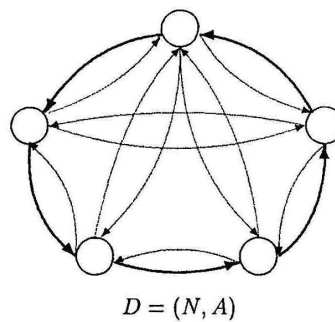


Figura 6: Esempio di circuito hamiltoniano

una sola volta tutti i vertici del grafo. La condizione necessaria (ma non sufficiente) per cui un grafo orientato G riesca a contenere un ciclo hamiltoniano implica che G debba essere connesso con $d_G(v) \geq 2$ per ogni $v \in V$; ovvero, ogni vertice deve avere come minimo grado 2, avendo così almeno 2 archi che vi si appoggiano, con almeno uno entrante ed almeno uno uscente.

Problemi polinomiali

Un problema si dice *polinomiale* se esiste un algoritmo polinomiale che lo risolve, ovvero un algoritmo che richiede, nel caso peggiore, un tempo di calcolo $f(d) = O(d^k)$, con d la dimensione dell'istanza risolta e k una costante fissata. Gli algoritmi polinomiali vengono considerati efficienti (indipendentemente dalla grandezza della costante k considerata).

Certificato polinomiale

La definizione della classe NP si basa quella di certificato polinomiale; viene definito *certificato polinomiale* una informazione ausiliaria che può essere utilizzata per verificare, in tempo polinomiale nella definizione dell'istanza, la correttezza della risposta data per una data istanza. Un problema viene quindi associato alla *classe NP* (Polinomiale Nondeterministico) se ogni sua istanza con risposta "si" ammette un certificato polinomiale, ovvero se fornita

Classe di problemi NP

una soluzione ad esso è possibile verificarne la correttezza. In particolare un problema si dice *NP-completo* se appartiene a NP e qualunque altro problema appartenente a NP è riconducibile ad esso in tempo polinomiale. Per i problemi appartenenti alla classe NP-difficile in particolare non è ancora noto un algoritmo polinomiale in grado di risolverli, ma non viene escluso che tale algoritmo esista.

NP-completezza

Il *Problema del Commesso Viaggiatore* effettua l'associazione a ciascuno degli

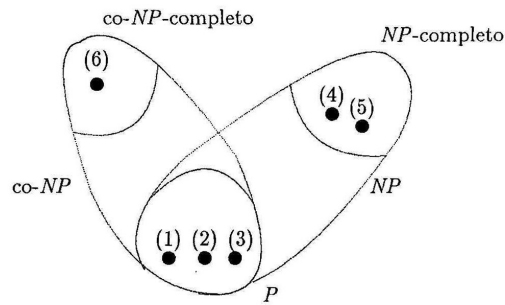


Figura 7: Classi di problemi

archi esistenti di un peso (che può anche variare a seconda della direzione dell'arco); l'algoritmo risolutivo di questo problema richiede la ricerca di un circuito in grado di passare per tutti i vertici (quindi hamiltoniano) del grafo orientato al quale viene applicato cercando di ottenere il costo minimo, che viene ottenuto come somma dei pesi singoli degli archi utilizzati per la sua composizione.

Questo problema fa parte della categoria dei problemi polinomiali, in particolare della classe di quelli NP-difficili e può essere descritto con un modello di programmazione lineare intera che utilizza le variabili decisionali x_{ij} , ciascuna delle quali rappresenta l'arco che parte da i e arriva in j e vale 1 se l'arco in questione viene scelto nel circuito ottimo, o altrimenti.

Il problema di PLI associato

$$x_{ij} = \begin{cases} 1 & \text{se l'arco } (i,j) \in A \text{ viene scelto nel circuito ottimo} \\ 0 & \text{altrimenti} \end{cases}$$

Si ottiene così che il TSP può essere scritto e risolto come il seguente problema di programmazione lineare intera:

$$\begin{aligned} & \min \underbrace{\sum_{(i,j) \in A} c_{ij} x_{ij}}_{\text{costo circuito}} \\ & \underbrace{\sum_{(i,j) \in \delta^-(j)} x_{ij} = 1, j \in V}_{\text{un arco entrante in } j} \\ & \underbrace{\sum_{(i,j) \in \delta^+(i)} x_{ij} = 1, i \in V}_{\text{un arco uscente da } i} \\ & \underbrace{\sum_{(i,j) \in \delta^+(S)} x_{ij} \geq 1, S \subset V : 1 \in S}_{\text{raggiungibilità da } 1} \\ & x_{ij} \geq 0 \text{ intero, } (i,j) \in A \end{aligned}$$

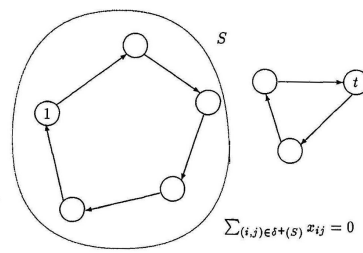


Figura 8: Soluzione applicabile al TSP senza definire la quarta riga dei vincoli

1. La *prima riga* indica che la soluzione ricercata consiste nel minimo valore presentato dalla funzione obiettivo, ovvero la sommatoria dei pesi degli archi che sono stati effettivamente selezionati come parte del cammino a costo minimo;
2. Dalla seconda alla quinta riga abbiamo la definizione dell'insieme delle soluzioni ammissibili; la *seconda riga* indica che per ogni vertice j deve essere scelto per la soluzione uno e un solo arco entrante; il circuito hamiltoniano per definizione passa una e una sola volta per ogni vertice;
3. Analogamente alla riga precedente la *terza riga* indica che per ogni vertice i deve essere effettuata la scelta di uno e un solo arco uscente da esso;
4. La *quarta riga* impone che la soluzione sia connessa, ovvero che ogni vertice $t \neq 1$ sia raggiungibile a partire dal vertice 1; vengono quindi esclusi i circuiti che non passano per il vertice 1, imponendo così che la soluzione consista di un unico circuito hamiltoniano;
5. L'*ultima riga* richiede la presenza di una soluzione intera (come definito precedentemente le variabili decisionali assumono unicamente i valori 0 e 1)

Il problema di programmazione lineare intera è quindi risolvibile, usando ad esempio il *metodo del semplice*.

Riduzione del problema pratico

Per poter applicare una soluzione analoga a quella del problema del commesso viaggiatore è necessario prima ricondursi ad essa dal caso pratico. Nel caso pratico consideriamo un grafo i cui vertici consistono nei luoghi di destinazione delle operazioni di disinfezione e gli archi definiti dal tragitto che si utilizza per raggiungere un luogo partendo da un'altro, che rappresentano tutte le possibili combinazioni se supponiamo che ogni luogo sia raggiungibile partendo da uno qualsiasi degli altri, e a ciascuno dei quali viene associato un peso pari alla distanza fisica tra i due luoghi. Avendo supposto il grafo come completo, ovvero l'esistenza di ogni arco che porta da un generico vertice v a un generico vertice w del grafo e viceversa, risulta verificata la condizione necessaria per l'esistenza di un circuito hamiltoniano nel grafo, ogni vertice dispone sicuramente di almeno un arco di ingresso ed almeno uno di uscita da esso; è possibile per gli stessi presupposti anche avere la certezza dell'esistenza di un circuito hamiltoniano; infatti è possibile ottenere un certificato polinomiale per tale affermazione collegando in un modo qualsiasi i vertici

del suddetto grafo. Il passo successivo consiste nel ricondurre il problema pratico al problema NP-completo del commesso viaggiatore.

Si dice che il problema P_1 si riduce in tempo polinomiale al problema P_2 ($P_1 \propto P_2$) se esiste un algoritmo per risolvere P_1 che utilizza come “procedura black-box” un qualunque algoritmo per P_2 e risulta polinomiale nell’ipotesi che l’algoritmo per P_2 richieda tempo $O(1)$. La procedura di ricerca del percorso minimo per le operazioni di disinfestazione è immediatamente riconducibile al problema del commesso viaggiatore; una volta ricevuti i dati dell’istanza del problema pratico, viene applicata una conversione logica che in tempo polinomiale porta ad avere un grafo orientato, che avviene in modo analogo a quello descritto precedentemente; l’istanza così ottenuta quindi può essere risolta con la stessa procedura utilizzabile ipoteticamente per risolvere il problema TSP.

Riduzione polinomiale

2.2.5 Gestione del Calendario

Il calendario costituisce una visualizzazione degli interventi programmati. La richiesta prevede che ci sia un calendario interventi aziendale, su base mensile oppure giornaliera, che consenta di visualizzare tutti gli interventi di tutti gli operatori per tutti i clienti.

2.3 COMPONENTI

A questo punto è possibile definire quali saranno le componenti hardware di cui dovrà disporre l’azienda committente per la messa in opera del sistema; Una delle richieste del committente consiste nell’accessibilità del servizio in ogni momento e da ogni luogo tramite web. Di conseguenza l’adempimento di questa richiesta richiede la presenza di un server interno all’azienda o alternativamente di un servizio di hosting web che garantisca la disponibilità e la raggiungibilità richieste. Inoltre per l’utilizzo del sistema, essendo previsto come modalità di utilizzo l’alternanza tra più di una funzionalità del sistema a brevi intervalli temporali è consigliabile l’utilizzo di due monitor per la visualizzazione, in modo da poter permettere l’uso e/o la visualizzazione di due schermate in contemporanea.

2.4 INTERFACCIA CON L’UTENTE

Come scritto precedentemente sono previste diverse tipologie di utenti finali: *amministratori del servizio, operatori addetti agli interventi e clienti dell’azienda*. E’ possibile notare quindi che in generale non si ha a che fare con utenti esperti, ma soprattutto con utenti occasionali e semi-occasionali (i clienti e gli operatori). Per questi utenti è previsto l’accesso a schermate specificatamente orientate a loro, ovvero la visualizzazione del calendario degli interventi relativi all’operatore singolo o per il cliente in questione; In particolare per queste schermate dunque si cercherà di avere un approccio in termini di interfaccia grafica il più intuitivo possibile. La realizzazione inoltre cercherà il più possibile di rispettare i requisiti software richiesti dallo standard ISO 9126, quali funzionalità, affidabilità, efficienza, usabilità, manutenibilità e portabilità.

3

STRUMENTI E TECNOLOGIE

Questo capitolo presenta i linguaggi, le librerie e gli strumenti utilizzati per realizzare il progetto, con i relativi vantaggi che vanno a motivare tali scelte.

INDICE

3.1	Linguaggi lato server	13
3.1.1	PHP	13
3.1.2	Doctrine ORM e DQL	14
3.1.3	MySQL e PHPMYAdmin	15
3.1.4	AJAX	15
3.2	Linguaggi lato client	15
3.2.1	HTML	15
3.2.2	CSS	17
3.2.3	Javascript	17
3.2.4	jQuery	18
3.2.5	jQuery UI	18

3.1 LINGUAGGI LATO SERVER

I linguaggi lato server sono quei linguaggi che vengono elaborati dal server, il quale mette a disposizione un insieme di servizi utili per il reperimento di informazioni o funzionalità non disponibili o di attuazione non raccomandabile nella macchina utente. Ad ogni richiesta effettuata dall'utente verrà poi spedito il risultato prodotto dal server.

3.1.1 PHP

PHP è un linguaggio di scripting interpretato, ha licenza open source e libera ed è concepito per la realizzazione di pagine web dinamiche. Attualmente è utilizzato principalmente per sviluppare applicazioni web lato server. L'elaborazione di codice PHP sul server produce codice HTML da inviare al browser dell'utente che ne fa richiesta. Il vantaggio maggiore dell'uso di PHP rispetto al classico HTML deriva dalla sua dinamicità nella creazione delle pagine che si contrappone alla staticità di quest'ultimo. PHP riprende per molti versi la sintassi del C e del PERL. È un linguaggio a tipizzazione debole e ha un buon supporto al paradigma di programmazione ad oggetti. Alcuni costrutti derivati dal C, come gli operatori fra bit e la gestione di stringhe come array, permettono in alcuni casi di agire a basso livello; tuttavia è fondamentalmente un linguaggio di alto livello, dotato di moltissime API e oltre 3000 funzioni nel nucleo base. PHP è in grado di interfacciarsi a innumerevoli database tra cui, nel caso particolare di questo progetto come illustrato in seguito, MySQL: è ottimamente integrato con esso e presenta una vasta quantità di script e librerie per interfacciarsi, tra cui menzionata in seguito la libreria Doctrine ORM. Si integra bene con altri linguaggi/piattaforme, moti-

<http://www.php.net>

Il passaggio dei parametri

vo per cui verrà fatto operare assieme a Javascript durante la realizzazione. Si è deciso di gestire il *passaggio di parametri* tra una pagina e l'altra soprattutto attraverso tre array di variabili PHP adibiti `$_GET`, `$_POST` e `$_SESSION`:

`$_GET` prevede il passaggio tramite la stringa che compare nella barra dell'indirizzo del browser

`$_POST` prevede un passaggio dei dati in background

`$_SESSION` rimane persistente durante una sessione intera e viene principalmente adibito alla gestione degli utenti

La preferenza nell'uso di PHP rispetto ad altri linguaggi lato server quali ASP è dovuta principalmente a tre fattori:

- Il modulo da realizzare dovrà riuscire a integrarsi bene con il sistema IES, il quale è realizzato anch'esso in PHP.
- La sua velocità di esecuzione delle istruzioni è superiore rispetto ad ASP richiedendo quindi meno tempo per il caricamento delle pagine
- la sua natura Open Source e compatibilità con pressochè ogni tipo di piattaforma server

3.1.2 Doctrine ORM e DQL

<http://www.doctrine-project.org>

Doctrine ORM è una libreria di PHP che fa parte del più grande Doctrine Project. Il Doctrine Project comprende un set di librerie PHP incentrate sul fornire servizi di persistenza dei dati e funzionalità correlate. ORM sta per Object Relational Mapper, ed è una libreria PHP che si occupa di applicare un livello di astrazione dal database (DBAL – DataBase Abstraction Layer). L'astrazione apportata consente una elevata portabilità del software utilizzato rispetto alla tecnologia del DBMS utilizzato; se si dovesse cambiare tecnologia al database non sarà necessario apportare alcuna modifica alle procedure scritte per la gestione e l'interrogazione della base di dati, basterà infatti modificare le righe del file di configurazione per indicare al software di riferirsi al nuovo DBMS. Una caratteristica rilevante di Doctrine è il basso livello di configurazione richiesto per cominciare un progetto. Doctrine può generare automaticamente classi di oggetti a partire da un database esistente, da lì in poi è possibile specificarvi relazioni e aggiungere funzionalità personalizzate; questa caratteristica permette di ottenere un codice ancora più leggibile, permettendo di trattare effettivamente a livello di codice le entità della base di dati come oggetti della programmazione orientata agli oggetti. Un'altra delle sue funzionalità chiave consiste nella possibilità di scrivere query sul database con un linguaggio proprietario orientato agli oggetti chiamato DQL (Doctrine Query Language). Questo meccanismo contribuisce ad aumentare il livello di astrazione dal database delle operazioni di accesso ai dati e migliora la leggibilità del codice. Inoltre Doctrine porta ulteriori vantaggi notabili come un sistema di cache per le query, che contribuisce a ridurre i tempi per reperire dati già richiesti precedentemente, o un sistema di aggregazione tra oggetti simili (che vengono poi distinti con l'aggiunta di una colonna tipo).

Doctrine Query Language

3.1.3 MySQL e PHPMyAdmin

MySQL è un Relation Database Management System (RDBMS); è Open Source, supporta la maggior parte della sintassi SQL ed è gestito da molti linguaggi di programmazione attraverso librerie, come Doctrine ORM. Sarà usato per l'impostazione e la gestione della struttura della base di dati con cui interagirà il modulo applicativo.

<http://www.mysql.com>

In particolare per velocizzare e semplificare tale procedura sarà utilizzato *PhpMyAdmin*. *PhpMyAdmin* è un'applicazione PHP gratuita che consente di amministrare in modo semplificato database di MySQL tramite un interfaccia web intuitiva. *PhpMyAdmin* supporta la maggior parte delle funzionalità di MySQL e quindi permette di creare un database da zero, creare le tabelle ed eseguire operazioni di ottimizzazione sulle stesse. Presenta un feedback sulla creazione delle tabelle per evitare eventuali errori. Sono previste delle funzionalità per l'inserimento dei dati (popolazione del database), per le query, per il backup dei dati, ecc.. L'applicazione può essere indirizzata sia agli amministratori del database, sia agli utenti.

<http://www.phpmyadmin.net>

3.1.4 AJAX

AJAX (Asynchronous Javascript and XML) è una tecnica di sviluppo per la realizzazione di applicazioni web interattive. Lo sviluppo di applicazioni con AJAX si basa su uno scambio di dati in background fra web browser e il server, in modo da consentire l'aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell'utente. AJAX è asincrono nel senso che i dati extra sono richiesti al server e caricati in background senza interferire con il caricamento della pagina esistente. Si decide di ricorrere a questa tecnica in contesti in cui è importante permettere all'utente di continuare ad essere operativo mentre il server elabora dati inviati precedentemente o per dividere le richieste dei dati essenziali da quelli facoltativi, in modo che possano essere caricati solo quando esplicitamente richiesti e così alleggerire (e quindi velocizzare) il caricamento iniziale di una pagina. L'uso di richieste AJAX sarà applicato tramite il linguaggio Javascript, menzionato in seguito.

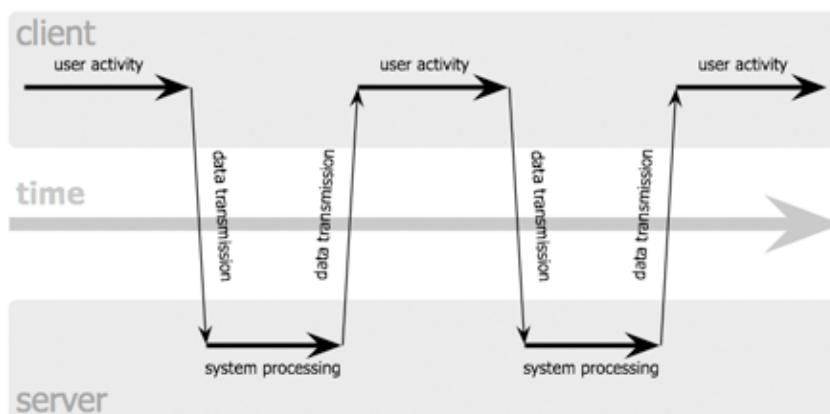
3.2 LINGUAGGI LATO CLIENT

I linguaggi lato client sono quei linguaggi la cui elaborazione e interpretazione vengono affidate alla macchina dell'utente (client). Di conseguenza una pagina web può essere visualizzata correttamente solo se il browser, con cui la si sta visitando, supporta tutte le tecnologie utilizzate. Per questo motivo si ricorrerà all'uso di linguaggi ad ampia compatibilità con i browser più diffusi; in particolare i linguaggi che saranno utilizzati per la gestione del lato client sono HTML, CSS, Javascript e le sue librerie jQuery e jQuery UI.

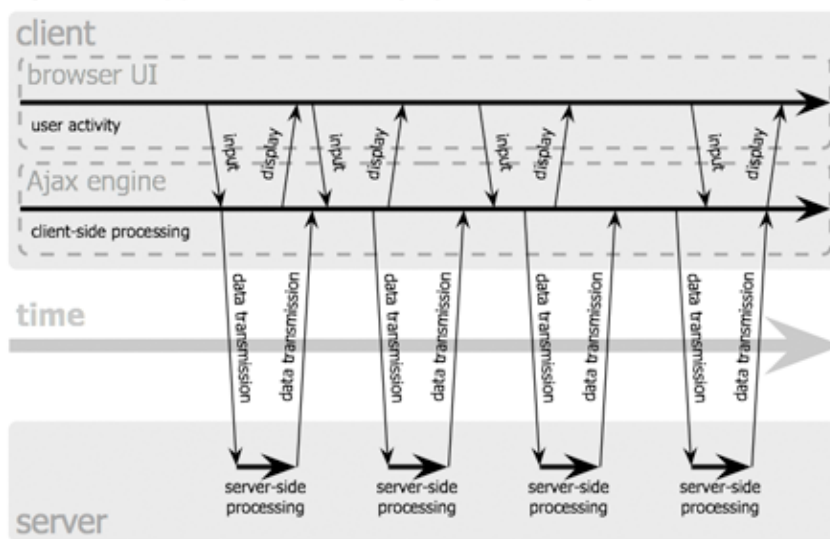
3.2.1 HTML

La sigla *HTML* è l'acronimo di HyperText Markup Language; è il linguaggio principalmente usato per definire i documenti ipertestuali disponibili nel World Wide Web. Esso non è un linguaggio di programmazione, bensì un linguaggio di markup, si occupa di descrivere le modalità di impaginazione,

classic web application model (synchronous)



Ajax web application model (asynchronous)



Jesse James Garrett / adaptivepath.com

Figura 9: Illustrazione del confronto tra modello sincrono (normale) e asincrono (con AJAX)

formattazione o layout del contenuto di una pagina web. Inoltre supporta l'inserimento di script esterni, come in questo particolare caso Javascript. Tuttavia HTML presenta due punti deboli che andranno risolti:

Punti deboli e loro soluzioni

- E' essenzialmente statico. I requisiti di dinamicità del modulo applicativo non permettono l'uso di html puro, quindi molto spesso si ricorre alla generazione dinamica del codice HTML tramite PHP.
- HTML è stato concepito principalmente per definire il contenuto logico e non l'aspetto finale del documento, per questo le sue tecniche di impostazione del layout sono molto spesso inefficaci nel fornire lo stesso risultato visivo su browser differenti; per questo motivo si gestirà l'impostazione dell'interfaccia grafica tramite i fogli di stile CSS.

3.2.2 CSS

Cascading Style Sheets è un linguaggio a fogli di stile usato per descrivere le semantiche di presentazione (ovvero l'aspetto e la formattazione) di un documento scritto con un linguaggio di markup (come HTML). Lo scopo primario di CSS è permettere la separazione tra contenuto del documento (scritto con HTML in questo caso) e sua presentazione, includendo elementi come il layout, colori e caratteri. Questa separazione è un grande vantaggio poiché:

- rende possibile avere un unico stile per pagine simili, riducendo così enormemente la lunghezza del codice;
- permette di differenziare la gestione della parte logica dell'applicativo da quella estetica, e quindi le loro relative manutenzioni.

CSS permette inoltre che lo stesso codice di markup sia presentato diversamente a seconda del contesto di visualizzazione, come ad esempio da browser, dispositivo mobile o per formato di stampa, consentendo così l'adattabilità dello stesso codice di markup ai diversi contesti di visualizzazione.

3.2.3 Javascript

Javascript è un linguaggio che definisce le funzionalità tipiche dei linguaggi di programmazione ad alto livello come strutture di controllo e cicli e consente l'utilizzo del paradigma orientato agli oggetti. Javascript è un linguaggio interpretato: il codice non viene compilato, ma interpretato in lato client, con l'interprete incluso nel browser utilizzato per la visualizzazione. Il vantaggio di questo meccanismo è che anche in presenza di script complessi il server non viene sovraccaricato a causa delle richieste dei client. Un suo svantaggio però è che uno script molto complesso può rallentare lo scaricamento portando a maggiori attese per l'esecuzione del codice. Inoltre ogni informazione che presuppone un accesso a dati memorizzati in un database remoto deve essere rimandata ad un linguaggio che effettui esplicitamente la transazione, per poi restituire i risultati ad una o più variabili Javascript; l'aggiornamento di tali informazioni richiede l'aggiornamento della pagina stessa. L'approccio che si intende usare per rimediare a questa situazione consiste nell'utilizzare PHP e Doctrine ORM per passare a Javascript i dati da database necessari per l'esecuzione del codice Javascript; nei contesti dove fosse necessario aggiornare questi dati evitando di aggiornare la pagina intera si ricorrerà all'uso della tecnica AJAX.

3.2.4 jQuery

<http://jquery.com>

jQuery è una libreria di funzioni per le pagine web, codificata in Javascript, che si propone come obiettivo quello di astrarre ad un livello più alto la programmazione lato client di ogni singola pagina HTML. E' in grado in questo modo anche di migliorare la compatibilità con browser diversi, rendendo le pagine visualizzabili e il codice javascript eseguibile in modo corretto dalla maggior parte dei browser di uso comune. Tramite l'uso della libreria *jQuery* è possibile effettuare operazioni mediamente complesse utilizzando poche righe di codice, come ottenere le dimensioni di un elemento e modificarle, oppure la standardizzazione e gestione automatica degli eventi. E' in grado anche di facilitare e standardizzare l'utilizzo di AJAX tramite funzioni utili e veloci che si occupano di istanziare i giusti oggetti ed effettuare la connessione e l'invio dei dati.

3.2.5 jQuery UI

<http://jqueryui.com>

jQuery UI (con UI che sta per User Interface) è una libreria complementare di *jQuery* che fornisce meccanismi di astrazione per interazioni e animazioni, effetti avanzati e widget personalizzabili. Il suo scopo è quello di creare applicazioni web altamente interattive e per questo il suo utilizzo pratico consisterà nella semplificazione dell'interfaccia grafica; nella gestione della programmazione interventi i principali cambiamenti orari potranno avvenire tramite semplici operazioni di trascinamento e ridimensionamento.

4 | REALIZZAZIONE

In questo capitolo sarà descritta l'effettiva realizzazione del software web, basata sulle analisi precedentemente effettuate e con gli strumenti precedentemente descritti. Come prima attività si progetta e crea la base di dati, per poi passare allo sviluppo delle varie funzionalità richieste.

INDICE

4.1	Progettazione della base di dati	19
4.1.1	Progettazione Concettuale	19
4.1.2	Progettazione Logica	25
4.1.3	Progettazione Fisica	26
4.1.4	Comunicazione con il modulo applicativo	27
4.2	Realizzazione del sistema di gestione degli Utenti e Login	28
4.3	Realizzazione del gestionale delle anagrafiche	29
4.4	Realizzazione del Gestionale delle Offerte e dei Contratti	31
4.5	Realizzazione della Schermata di Gestione del Calendario	36
4.6	Realizzazione della Programmazione degli Interventi	37
4.6.1	Schedulazione automatica degli interventi	38
4.6.2	Schedulazione manuale	52

4.1 PROGETTAZIONE DELLA BASE DI DATI

Il primo passo nella realizzazione del modulo applicativo consiste nella progettazione e implementazione di una base di dati sul quale esso andrà ad appoggiarsi, le specifiche richieste infatti rendono essenziale la presenza di un sistema di persistenza dei dati efficace ed efficiente.

4.1.1 Progettazione Concettuale

Nel capitolo 2 sono stati individuate e studiate le proprietà e le funzionalità richieste dal sistema; sulla base di questa analisi dei requisiti si procede nella progettazione della base di dati sul quale il modulo applicativo andrà ad appoggiarsi attraverso una strategia di progettazione di tipo inside-out; si comincia dunque con il delineare inizialmente i concetti importanti e si procede estendendosi a macchia d'olio da questi verso quelli concettualmente più lontani attraverso una navigazione tra le specifiche.

I termini chiave evidenziati nella descrizione della problematica sono riportati nella tabella 1 nella pagina seguente, con la loro descrizione.

Tabella 1: Dizionario dei dati

Concetto	Attributi	Descrizione
Contatto	Nome, cognome, ragione sociale, telefono, fax, email, partita iva, codice fiscale, indirizzo fatturazione, email ufficio acquisti, email ufficio vendite	Singolo o azienda che cerca di stabilire relazioni contrattuali con il committente. Può chiedere la disinfestazione di più ambienti o da più di un infestante.
Cliente	Nome, cognome, ragione sociale, telefono, fax, email, partita iva, codice fiscale, indirizzo fatturazione, email ufficio acquisti, email ufficio vendite	Singolo o azienda che ha stabilito relazioni contrattuali con il committente. Può chiedere la disinfestazione di più ambienti o da più di un infestante.
Luogo	Comune, indirizzo, numero civico, nominativo, telefono, riferimento, latitudine, longitudine	Destinazione fisica di uno o più interventi di disinfestazione legata ad un cliente/contatto.
Operatore	Nome, cognome, data nascita, telefono fisso, telefono mobile, telefono azienda, email, indirizzo residenza, nazionalità, stagionale	Persona che fa da tramite per l'azienda per l'effettuazione degli interventi.
Infestante	Nome, inizio periodo infestazione, fine periodo infestazione, foto	Animale invasivo sul quale è possibile intervenire.
Mezzo	Nome, targa, telaio, data acquisto, data immatricolazione, numero polizza, scadenza assicurazione, cassetta primo soccorso, scadenza cassetta, scadenza bollino blu, scadenza bollo, scadenza revisione	Mezzo utilizzabile dagli operatori per gli spostamenti.
Attrezzatura	Nome, numero matricola, data acquisto, data scadenza revisione	Materiale tecnico usato dagli operatori per effettuare gli interventi.
Prodotto	Nome, scheda sicurezza, scheda tecnica, principio attivo	Materiale consumabile utilizzato nelle operazioni di disinfestazione.
Vincolo		Condizione che limita l'intervento degli operatori nell'effettuare la disinfestazione.
Vincolo generico	Descrizione	Condizione generica che riguarda ad esempio la presenza di un cane, di una persona disabile, ecc.

Tabella 1: Continua nella prossima pagina

Tabella 1: Continua dalla pagina precedente

Concetto	Attributi	Descrizione
Vincolo accessibilità giornaliero	Data	Condizione che limita i giorni in cui sono effettuabili gli interventi in un luogo.
Vincolo accessibilità orario	Ora	Condizione che limita gli orari in cui sono effettuabili gli interventi in un luogo.
Vincolo Post-intervento	Positivo	Condizione derivante da un intervento precedente in una serie di interventi.
Vincolo di destinazione	Descrizione	Condizione dipendente dal luogo in cui viene effettuato l'intervento, ad esempio il piano del palazzo, la presenza di un cancello, ecc.
Zona	Nome	Suddivisione, solitamente spaziale, del territorio atta a raggruppare i luoghi di intervento.
Tipologia di intervento	Nome, descrizione, numero interventi, intervallo interventi, durata intervento	Procedura effettivamente attuabile dall'azienda. Va ad operare su un Infestante e può richiedere delle attrezzature specifiche.
Fornitore	Nome, email ufficio acquisti, email ufficio vendite, coordinate bancarie	Ente che fornisce all'azienda il materiale a sua disposizione (mezzi, attrezzature, prodotti).
Offerta	Codice, data inizio, data fine validità, preventivo	Comprende un insieme di dati relativi ad un caso di disinfestazione quali generalità del cliente, luoghi da disinfestare e tipologie di intervento da eseguire, ed un preventivo che deve essere valutato dal cliente.
Contratto	Codice, tipo contratto, data stipulazione, data scadenza, costo, tacito rinnovo	Comprende un insieme di dati relativi ad un caso di disinfestazione quali generalità del cliente, luoghi da disinfestare e tipologie di intervento da eseguire, ed un costo effettivo che è stato accettato dal cliente.

Tabella 1: Continua nella prossima pagina

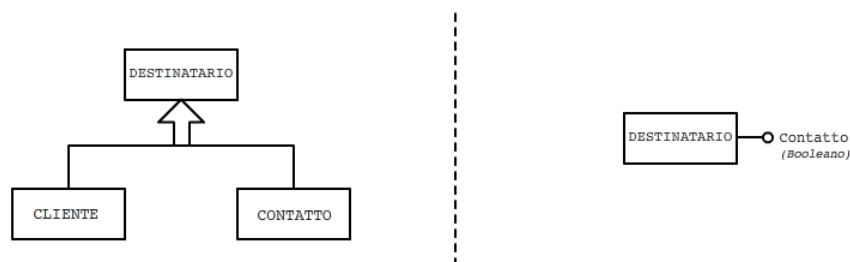


Figura 10: Le entità contatto e cliente a livello concettuale prima e dopo la ristrutturazione

Tabella 1: Continua dalla pagina precedente

Concetto	Attributi	Descrizione
Sessione	Numero interventi, intervallo interventi, durata intervento, inizio sessione, fine sessione	Rappresenta un insieme di operazioni di disinfestazione. Un contratto viene composto da una o più sessioni. Ciascuna sessione è identificata univocamente dal Contratto di cui fa parte, dal Luogo su cui va operare e dalla Tipologia di Intervento a cui si dedica.
Istanza di intervento	Data, ora, durata, stato, report, successo, catture	Rappresenta l'intervento effettivamente eseguito da un insieme di operatori, mezzi e attrezzature in una sessione. Una sessione è composta da una o più istanze di intervento.

Tabella 1: Si conclude dalla pagina precedente

Prima di procedere alla stesura di un modello concettuale della base di dati è necessario fare alcune valutazioni ed eseguire di conseguenza delle ristrutturazioni per ottimizzare lo schema e semplificarne la realizzazione a livello logico successivamente:

Accorpamento di entità 1

Contatto e Cliente sono due entità estremamente simili tra loro al punto che possono essere generalizzate come una stessa entità, quindi si decide di effettuarne un accorpamento; si provvede ad aggiungere un campo flag per distinguere i casi.

Accorpamento di entità 2

Si applica un ragionamento analogo a Contratto e Offerta e si procede anche in questo caso con una operazione di accorpamento; il campo costo potrà valere come campo preventivo (riducendo quindi i valori nulli di tale accorpamento) e il campo tipo contratto presenterà uno stato extra per indicare lo stato di offerta;

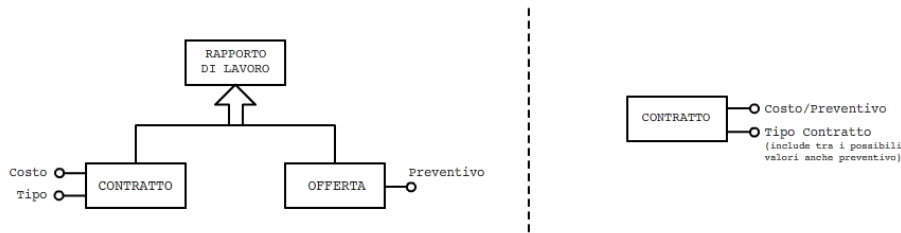


Figura 11: Le entità contratto e offerta a livello concettuale prima e dopo la ristrutturazione

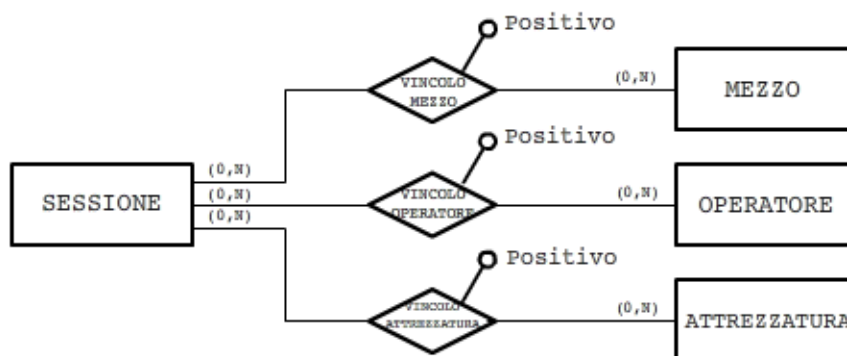


Figura 12: Schema concettuale dei vincoli post-intervento

Non hanno avuto un vasto approfondimento nella analisi dei requisiti iniziale, tuttavia è fondamentale l'introduzione dei due concetti di Sessione e Istanza di Intervento, che saranno discussi più approfonditamente nei successivi paragrafi;

E' stata ritenuta ottimale l'aggiunta dell'attributo ridondante "interventi effettuati" sull'entità sessione che fa riferimento al numero di istanze di intervento completate con successo che vi sono collegate; tale ridondanza riduce sostanzialmente la complessità delle operazioni che fanno accesso a questo dato, pur occupando uno spazio in memoria extra di grandezza trascurabile;

Ridondanza su Sessione

Nonostante siano presenti quattro occorrenze nominate come Vincolo stiamo parlando di concetti abbastanza diversi tra di loro come, quindi non generalizzabili come una unica entità Vincolo; in particolare Vincolo Generico sono concetti relativi all'entità istanza di intervento, Vincoli Post-intervento sono relativi alla Sessione mentre Vincolo Temporale e Vincolo di destinazione sono relativi a Luogo;

L'entità Vincolo

I Vincoli Post-intervento indicano che da un momento temporale in poi una sessione debba (o non debba) associare alle sue Istanze di intervento uno o più determinati Operatori, Mezzi o Attrezzature. Il campo Positivo è un booleano che indica l'obbligatorietà o alternativamente la negazione di tali legami a livello di Istanza;

I Vincoli Post-intervento

Lo schema E-R ricavato e ristrutturato è dunque il seguente (per semplicità gli attributi non identificativi e lo schema di figura 12 sono stati tralasciati):

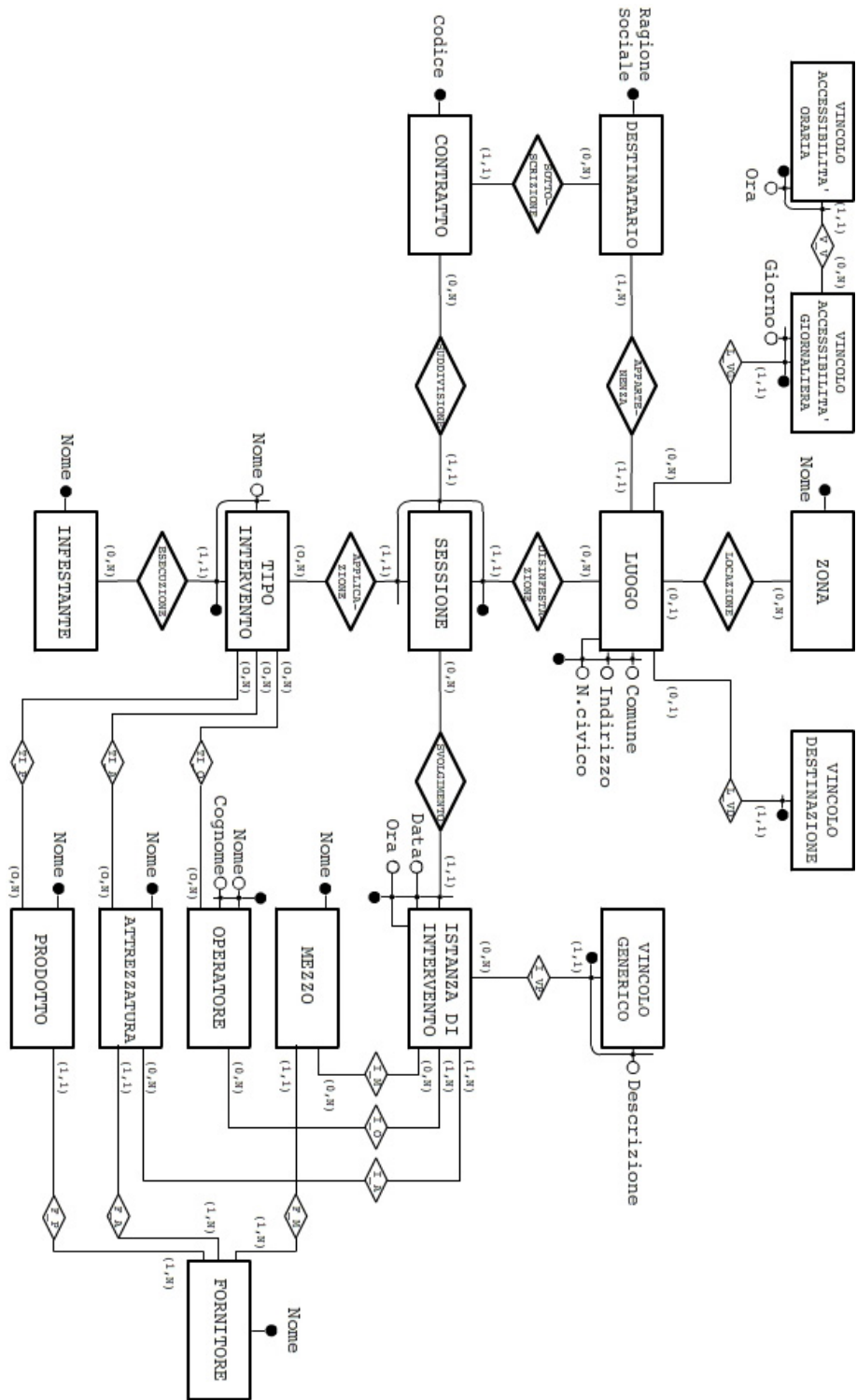


Figura 13: Schema Entità-Associazione della base di dati

4.1.2 Progettazione Logica

A questo punto si passa alla progettazione logica; lo scopo è quello di costruire, basandosi sulle informazioni contenute nello schema Entità Associazione creato precedentemente, uno schema logico in grado di descriverlo in maniera corretta ed efficiente. E' stata effettuata una ulteriore modifica rispetto allo schema precedente nel passaggio, ovvero l'aggiunta di un identificatore per ogni tabella; tale modifica ha lo scopo di facilitare l'integrazione con IES e semplificare le tabelle dotate di chiavi esterne. Inoltre per aumentare la semplicità di gestione dei luoghi nella schermata delle anagrafiche (approfondita nei paragrafi successivi) sono state aggiunte le tre tabelle Comune, Provincia e Regione.

Aggiunta di identificatori univoci

Aggiunta di Comune, Provincia e Regione

Lo schema logico-relazionale ricavato è il seguente:

Destinatario (Id, Nome, Cognome, Ragione sociale, Telefono, Fax, Email, Partita iva, Codice fiscale, Indirizzo fatturazione, Email ufficio acquisti, Email ufficio vendite, Contatto)

Contratto (Id, Codice, Tipo contratto, Data inizio offerta, Data fine offerta, Data stipulazione, Data scadenza, Costo, Tacito rinnovo, Destinatario)

Luogo (Id, Comune, Indirizzo, Numero civico, Nominativo, Telefono, Riferimento, Latitudine, Longitudine, Destinatario, Zona, Vincolo di destinazione)

Comune (Id, Nome, CAP, Provincia)

Provincia (Id, Nome, Sigla, Regione)

Regione (Id, Nome)

Sessione (Id, Interventi effettuati, Numero interventi, Intervallo interventi, Durata intervento, Inizio sessione, Fine sessione, Contratto, Luogo, Tipologia Intervento)

Istanza (Id, Data, Ora, Durata, Stato, Report, Successo, Catture, Sessione)

Vincolo_generico (Id, Descrizione, Istanza)

Mezzo (Id, Nome, Targa, Telaio, Data acquisto, Data immatricolazione, Numero polizza, Scadenza assicurazione, Cassetta primo soccorso, Scadenza cassetta, Scadenza bollino blu, Scadenza bollo, Scadenza revisione, Fornitore)

Operatore (Id, Nome, Cognome, Data nascita, Telefono fisso, Telefono mobile, Telefono azienda, Email, Indirizzo residenza, Nazionalità, Stagionale)

Attrezzatura (Id, Nome, Numero matricola, Data acquisto, Data scadenza revisione, Fornitore)

Fornitore (Id, Nome, Email ufficio acquisti, Email ufficio vendite, Coordinate bancarie)

Prodotto (Id, Nome, Scheda sicurezza, Scheda tecnica, Principio attivo, Fornitore)

Tipologia intervento (Id, Nome, Descrizione, Numero interventi, Intervallo interventi, Durata intervento, Infestante)

Infestante (Id, Nome, Inizio periodo infestazione, Fine periodo infestazione, Foto)

Zona (Id, Nome)

Vincolo accessibilita giornaliera (Id, Giorno, Luogo)

Vincolo accessibilita oraria (Id, Ora, Vincolo accessibilita giornaliera)

Vincolo mezzo (Id, Sessione, Mezzo, Positivo)

Vincolo operatore (Id, Sessione, Operatore, Positivo)

Vincolo attrezzatura (Id, Sessione, Attrezzatura, Positivo)

I.M (Id, Istanza, Mezzo)

I.O (Id, Istanza, Operatore)

I.A (Id, Istanza, Attrezzatura)

4.1.3 Progettazione Fisica

In questa fase lo schema logico viene completato con la specifica dei parametri fisici di memorizzazione dei dati. L'implementazione fisica vera e propria della base di dati avverrà, come anticipato in precedenza, tramite l'uso del gestionale PhpMyAdmin, che provvederà a generare il codice MySQL necessario alla definizione della base di dati.

Di seguito è fornito un esempio del codice MySQL generato per la creazione di alcune delle principali tabelle:

```
CREATE TABLE CONTRATTO (
  ID INT,
  CODICE INT,
  TIPO_CONTRATTO VARCHAR(45) NOT NULL,
  DATA_INIZIO_OFFERTA TIMESTAMP,
  DATA_FINE_OFFERTA TIMESTAMP,
  DATA_STIPULAZIONE TIMESTAMP,
  DATA_SCADENZA TIMESTAMP,
  COSTO FLOAT,
  TACITO_RINNOVO BOOLEAN,
  DESTINATARIO INT NOT NULL,
  PRIMARY KEY(ID),
  FOREIGN KEY(DESTINATARIO) REFERENCES DESTINATARIO(ID)
);
```

```

CREATE TABLE LUOGO (
  ID INT,
  COMUNE INT NOT NULL,
  INDIRIZZO VARCHAR(255) NOT NULL,
  NUMERO_CIVICO VARCHAR(7) NOT NULL,
  NOMINATIVO VARCHAR(255),
  TELEFONO VARCHAR(45),
  RIFERIMENTO VARCHAR(45),
  LATITUDINE FLOAT,
  LONGITUDINE FLOAT,
  DESTINATARIO INT NOT NULL,
  ZONA INT,
  VINCOLO_DESTINAZIONE VARCHAR(1023),
  PRIMARY KEY(ID),
  FOREIGN KEY(DESTINATARIO) REFERENCES DESTINATARIO(ID),
  FOREIGN KEY(ZONA) REFERENCES ZONA(ID)
);

CREATE TABLE ISTANZA (
  ID INT,
  DATA DATE NOT NULL,
  ORA TIME NOT NULL,
  DURATA FLOAT NOT NULL,
  STATO VARCHAR(7) NOT NULL,
  REPORT VARCHAR(255),
  SUCCESSO BOOLEAN,
  CATTURE INT,
  SESSIONE INT NOT NULL,
  PRIMARY KEY(ID),
  FOREIGN KEY(SESSIONE) REFERENCES SESSIONE(ID)
);

```

4.1.4 Comunicazione con il modulo applicativo

Una volta terminata la realizzazione fisica della base di dati non resta che effettuare il collegamento con il software web in modo che sia possibile effettuare agevolmente tramite codice le operazioni di lettura e scrittura che saranno necessarie durante la sua messa in opera. Tale collegamento viene effettuato tramite il file di implementazione di Doctrine ORM; Questo file a partire dai dati del database provvederà a creare automaticamente un modello a oggetti delle entità contenute al suo interno, in modo poi da semplificarne la gestione a livello di codice. Di seguito è presentato il codice che verrà utilizzato dal file di implementazione delle classi di Doctrine ORM:

```

<?php
/* bootstrap.php */

require_once(dirname(__FILE__) . '/Doctrine/lib/Doctrine.php');

$client_root = "nome_root";
$client = "nome_client";
$db_host = "192.168.1.1";

```

```

$db_user = "nome_utente_db";
$db_password = "password_utente_db";
$db_name = "nome_db";

spl_autoload_register(array('Doctrine', 'autoload'));
//spl_autoload_register(array('Doctrine', 'modelsAutoload'));

$Manager = Doctrine_Manager::getInstance();

// Questo codice rende possibile l'override dei setter/getter
/*
$Manager->setAttribute(Doctrine_Core::ATTR_AUTO_ACCESSOR_OVERRIDE,
    true);
$Manager->setAttribute(Doctrine_Core::ATTR_MODEL_LOADING,Doctrine_Core::
    MODEL_LOADING_CONSERVATIVE);
$Manager->setAttribute(Doctrine_Core::ATTR_AUTOLOAD_TABLE_CLASSES,
    TRUE);
*/

$Manager->setAttribute(Doctrine_Core::ATTR_AUTO_ACCESSOR_OVERRIDE,
    TRUE);

$Manager->setCharset('utf8');
$Manager->setCollate('utf8_unicode_ci');

// Connessione
$conn = Doctrine_Manager::connection("mysql://$db_user:
    $db_password@$db_host/$db_name",$db_name");

// Generazione modelli
Doctrine_Core::generateModelsFromDb('new_models', array($db_name), array('
    generateTableClasses' => true));

// Autoload dei models
Doctrine_Core::loadModels(array(dirname(__FILE__).'/../models/generated',
    dirname(__FILE__).'/../models'));
?>

```

4.2 REALIZZAZIONE DEL SISTEMA DI GESTIONE DEGLI UTENTI E LOGIN

Le richieste dell'analisi iniziale prevedono una gestione del sistema orientata agli utenti, con uno o più amministratori in grado di creare nuovi account su necessità; inoltre è necessario l'uso di una pagina di login per consentire l'uso solo agli utenti registrati. Tali funzionalità tuttavia non necessitano alcuna implementazione in questo modulo applicativo, poiché sono già gestite dal sistema IES; Ciò che invece dovrà essere gestito sono le funzionalità particolari che possono variare a seconda dei permessi concessi alle varie tipologie di utente, che saranno trattate nei paragrafi seguenti.

4.3 REALIZZAZIONE DEL GESTIONALE DELLE ANAGRAFICHE

La seconda funzionalità che va trattata secondo le richieste è la Gestione delle Anagrafiche. Lo scopo di questa sezione è permettere all'amministratore del sistema di visionare e gestire i dati contenuti nella base di dati pur mantenendo un discreto livello di astrazione concettuale da essi. Questo permette all'utente di poter utilizzare al meglio il sistema pur senza avere bisogno di conoscere la struttura della base di dati su cui si appoggia. Le operazioni di modifica semplici sulle tabelle vengono inoltre incapsulate in operazioni di alto livello; questo serve per fare in modo che i vincoli di chiave esterna, di non nullità e comunque ogni vincolo che richiede definizione sul codice non rischino di essere violati.

Le entità per le quali è richiesta una gestione di questo tipo, come richiesto dall'analisi iniziale, sono le seguenti:

- Attrezzature
- Clienti/Contatti
- Fornitori
- Infestanti
- Luoghi
- Mezzi
- Operatori
- Prodotti
- Tipologie di Intervento

Sono qui definite cinque funzionalità di base che vanno ad operare su ciascuna anagrafica:

- Ricerca di uno o più elementi
- Visualizzazione di un elemento
- Modifica di un elemento
- Eliminazione di un elemento
- Creazione di un elemento

Queste funzionalità avranno la possibilità di essere regolate secondo l'anagrafica a cui si riferiscono (ad esempio non sempre può essere possibile eliminare un elemento) oppure secondo la tipologia di utente che vi accede (alcuni utenti potrebbero avere unicamente permessi di visualizzazione). La struttura quindi della parte del sistema che gestisce le anagrafiche viene quindi strutturata nel modo seguente:

Ogni anagrafica si presenta con un elenco di tutte le righe che l'utente in visualizzazione è abilitato a vedere; a questo punto possono essere proposte le cinque azioni citate precedentemente; in particolare le operazioni di

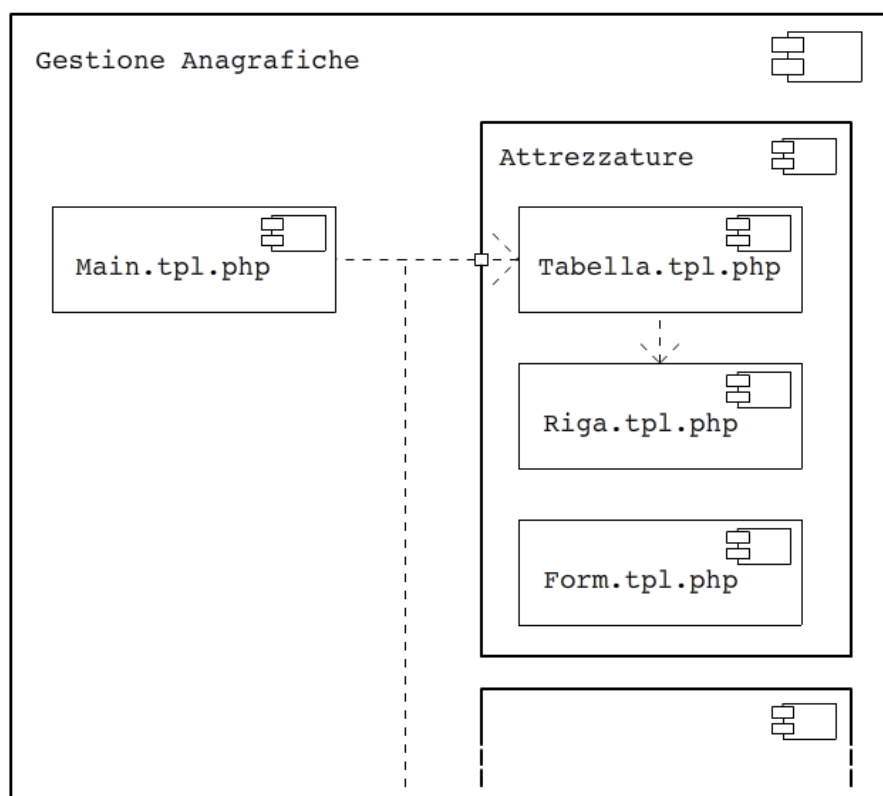


Figura 14: Diagramma dei Componenti della Gestione Anagrafiche

modifica e creazione comporteranno la proposta di un modulo da compilare con i dati che si intendono aggiungere/modificare e quelle di visualizzazione comporteranno invece l'apparizione di un modulo con i campi dettagliati relativi all'elemento di cui si effettua la visualizzazione. Come da figura 14, si può notare che i file delegati alla gestione di ciascuna anagrafica sono principalmente tre:

TABELLA.TPL.PHP La lista degli elementi viene caricata qui, tramite l'inclusione del file `Riga.tpl.php` per ciascuna riga. Inoltre sono da qui accessibili le operazioni che riguardano gli elementi nel loro insieme o nessun elemento preesistente, quali la ricerca e la creazione di nuovi elementi.

RIGA.TPL.PHP E' il file relativo alla singola riga della tabella degli elementi. Ogni riga presenta gli attributi più rilevanti dell'elemento, sui quale è possibile applicare filtri di ricerca, e le tre operazioni associabili a ciascun elemento esistente, quali la visualizzazione, la modifica e l'eliminazione.

FORM.TPL.PHP Come da nome, è composto da un modulo formato dall'insieme delle proprietà di un determinato elemento e da una lista degli elementi che vi sono associati, che eventualmente sono entrambi modificabili. Questo modulo viene richiamato tramite i file precedentemente descritti per effettuare le operazioni di modifica, visualizzazione e creazione di un elemento; nel caso delle prime due i campi risultano precompilati con i valori precedentemente attribuiti. Il caricamento del form viene praticato tramite la procedura dialog della libreria jQuery UI; tramite questa procedura è possibile caricare la pagina in modo simile all'apertura di pop-up senza dover quindi aggiornare la pagina di provenienza, potendo quindi ritornarvi più agevolmente. Inoltre l'operazione di invio dei dati, che termina rispettivamente la modifica e la creazione di un elemento, è effettuata tramite procedura AJAX. Infine, per effettuare pienamente il concetto di incapsulamento delle operazioni di modifica sul dato, la pressione del tasto di ok prima di effettuare l'invio dei dati provvede a controllare i campi degli attributi per accertare che i valori inseriti rispettino i vincoli di chiave, non nullità, di dominio ed eventualmente i vincoli che per ragioni particolari non sono stati esprimibili direttamente sulla base di dati; se uno o più di questi vincoli vengono violati l'invio viene bloccato ed i campi in questione segnalati visivamente all'utente, in modo che possa provvedere a correggerli prima di effettuarne nuovamente l'invio.

4.4 REALIZZAZIONE DEL GESTIONALE DELLE OFFERTE E DEI CONTRATTI

La schermata di gestione di offerte/contratti è uno dei punti chiave del sistema software, dovuto al fatto che si ha qui la base per la schedulazione successiva degli interventi; per questo vi è stato dedicato un paragrafo alla sua progettazione.

La sua struttura di base è simile a quella delle anagrafiche descritta precedentemente, visibile dallo schema di figura 14; tuttavia la schermata del form è fondamentale per questa gestione e verrà descritta più approfonditamente.

Dalla discussione con il committente si deduce che lo specifico modo in cui si arriverà alla creazione dei contratti è il seguente:

SCENARIO BASE:

- 1 Il destinatario contatta telefonicamente l'azienda per richiedere un servizio di disinfestazione
- 2 Il gestore del servizio avvia una proposta di contratto (offerta)
- 3 Il destinatario è già presente nel database dei clienti, quindi viene aggiunto all'offerta
- 4 Il destinatario comunica per quali dei luoghi precedentemente assegnatagli intende eseguire le operazioni di disinfestazione
- 5 Il gestore riceve i dettagli del problema di infestazione da parte del cliente
- 6 Il gestore comunica le tipologie di intervento che necessitano di essere eseguite per porre fine alla problematica
- 7 Il gestore formula un preventivo e fissa una data di fine validità all'offerta di contratto per l'utente
- 8 Il destinatario accetta l'offerta
- 9 L'offerta diventa contratto, viene fissata una scadenza ed entra in opera

ALTERNATIVE:

- 3.A Il destinatario è un nuovo contatto dell'azienda, quindi vengono raccolti i suoi dati
- 4.A I luoghi per il quale il cliente intende eseguire le operazioni non sono ancora registrati nella base di dati, quindi ne vengono raccolti i dati relativi.
- 6.A La problematica non è ancora pienamente identificata; un operatore viene inviato sul posto per accertare la situazione
- 7.A Il destinatario richiede un servizio a breve termine, quindi il gestore cerca un momento temporale disponibile sul calendario degli interventi e l'offerta diventa contratto a singolo intervento (contratto Spot), concludendo il caso d'uso
- 8.A Il destinatario contratta sui dettagli dell'offerta
- 8.B Il destinatario rifiuta l'offerta, che viene annullata (il caso d'uso si conclude)

Da qui è possibile inoltre riconoscere l'esistenza di un insieme di operazioni fondamentali che richiedono di essere gestite per il corretto funzionamento, ovvero nel dettaglio:

- la possibilità del cliente di contattare l'azienda per richiedere un servizio di disinfestazione

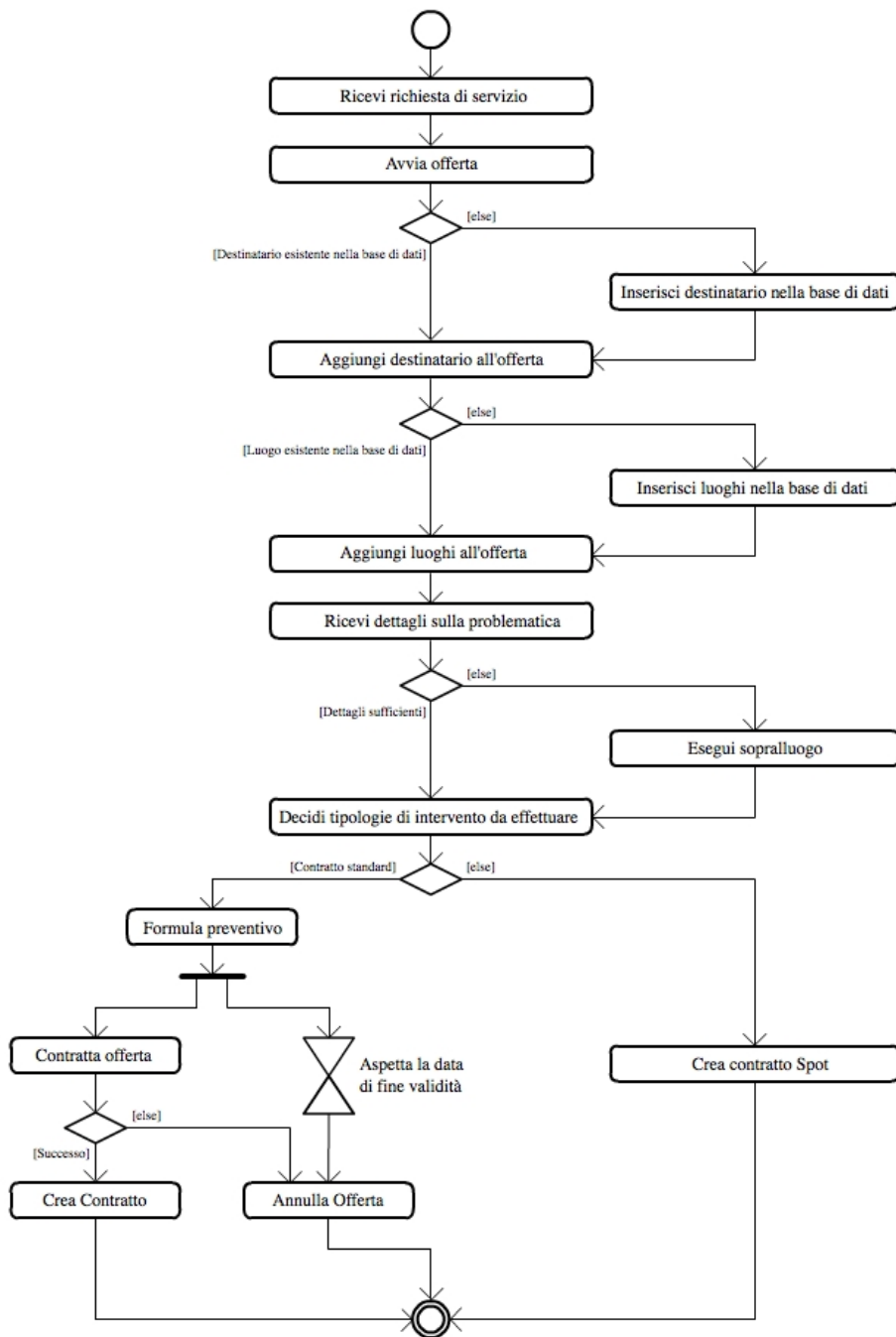


Figura 15: Diagramma delle attività riguardanti la creazione dei contratti

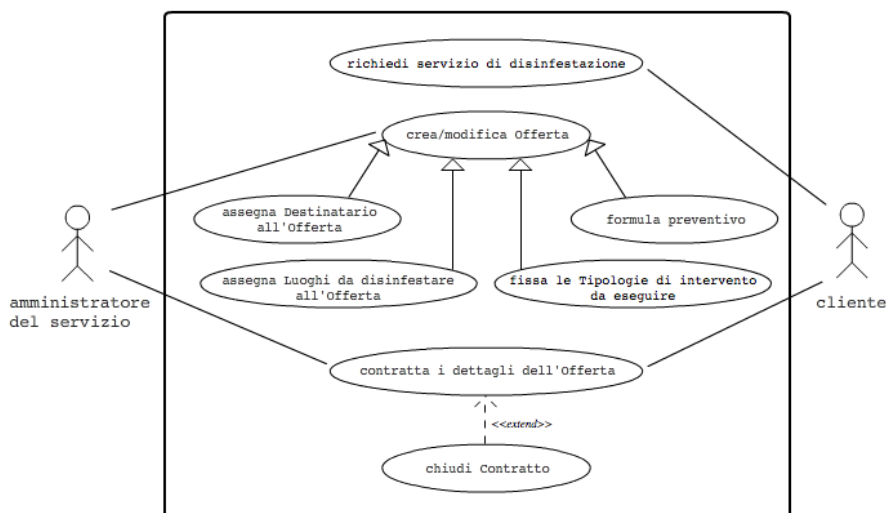


Figura 16: Diagramma dei casi d'uso riguardante lo scenario base nella creazione dei contratti

- la possibilità dell'amministratore di creare un'offerta o alternativamente modificarne una già esistente, in particolare per sceglierne il destinatario, assegnare uno o più luoghi di destinazione delle operazioni di disinfestazione, decidere le tipologie di intervento da eseguire e formulare un preventivo o variarne uno precedentemente esistente
- la possibilità di avere una fase di contrattazione tra le due parti che consenta di arrivare ad un accordo sui dettagli dell'offerta e quindi al suo passaggio di stato a Contratto effettivo

Quindi, riferendosi a questo modello, si decide di procedere nella realizzazione del modulo di gestione delle offerte nel modo seguente:

Sezione di scelta Destinatari

- E' prevista una sottosezione per la scelta del destinatario. Sarà fornita una modalità di selezione a partire da una lista dei Destinatari precedentemente esistenti nella base di dati. Qualora capitasse il caso in cui il Destinatario in questione fosse un nuovo contatto per l'azienda (alternativa 3.a) sarà offerta la possibilità di effettuare la sua inserzione direttamente nella base di dati, proponendo un sotto-modulo con campi relativi all'inserzione dei destinatari.

Sezione di scelta Luoghi

- E' prevista una sottosezione in modo analogo anche per la scelta dei luoghi per cui sono richieste le operazioni di disinfestazione. Anche qui sarà presente una lista dei luoghi precedentemente esistenti per permetterne una scelta veloce. Una peculiarità aggiuntiva che si è scelto di implementare è la proposta unicamente dei luoghi direttamente collegati al Destinatario selezionato precedentemente. Questo è dovuto al fatto che è stato definito un vincolo sul codice che permette ad un'offerta di effettuare la disinfestazione unicamente sui luoghi collegati al proprio destinatario. Questa scelta progettuale permette di velocizzare l'operazione di scelta. Bisogna comunque notare che nel caso nella fase precedente fosse stato rilevato un contatto nuovo per l'azienda non saranno

comunque presenti luoghi precedentemente esistenti da suggerire per essere selezionati. Sempre analogamente alla sezione precedente sarà resa disponibile la possibilità di aggiungere un numero arbitrario di moduli supplementari per l'inserimento dei nuovi luoghi, ciascuno dotato dei campi necessari per l'inserimento, per consentire l'immissione di luoghi non precedentemente esistenti nella base di dati (alternativa 4.a). La natura di queste due modalità di aggiunta dei luoghi è in grado di permettere l'inserimento anche alternato di luoghi preesistenti con quelli appena aggiunti.

- Sarà realizzata una sezione dedicata alla specifica delle tipologie di intervento, in cui sarà possibile stabilire le modalità e le caratteristiche con cui verrà applicata la soluzione alla problematica. Ad ogni luogo selezionato nella fase precedente, per il quale è stata appunto richiesta la procedura, sarà possibile associare una o più tipologie di intervento da effettuare, selezionabili tra quelle precedentemente presenti all'interno della base di dati e distinte per animale infestante al quale si riferiscono, in modo da permettere allo stesso tempo la coesistenza di tipologie di intervento omonime per infestanti diversi e di risalire facilmente al nome di una procedura partendo dalla conoscenza dell'infestante al quale è associata. Ci troviamo quindi ad avere una serie di associazioni tra luoghi e tipologie di intervento che vi vengono effettuate, che diventeranno le cosiddette Sessioni che abbiamo definito a livello concettuale. Per ogni sessione definita quindi sarà possibile immettere un modulo con i dettagli relativi ad essa, i quali nel particolare come definito precedentemente sulla base di dati sono: numero degli interventi da effettuare, intervallo in giorni che deve trascorrere tra un intervento ed il successivo, durata del singolo intervento, date di inizio e fine della sessione; per velocizzare l'immissione dei primi tre, saranno suggeriti tre valori standard predefiniti durante la fase di immissione della tipologia di intervento alla quale la sessione si riferisce; tuttavia ciascuno di essi è personalizzabile per ogni singola sessione. Oltre a questi parametri viene reso possibile definire a livello di sessione i Vincoli post-intervento, che data la loro particolare natura saranno modificabili anche quando l'offerta sarà diventata contratto. *Sezione delle Tipologie di Intervento*
- Viene data la possibilità di modificare i parametri strettamente legati all'offerta stessa che sono ragionati in base al servizio che è stato precedentemente definito; è possibile definire un preventivo da proporre al cliente per la messa in opera del contratto e una data di fine validità per tale offerta, allo scadere della quale la porterà all'annullamento. *Modifica Preventivo e scadenze*
- In casi particolari viene resa possibile la modifica dello stato dell'offerta, ovvero riferendosi allo schema di figura 4 a pagina 6 relativo agli stati di un contratto nel dettaglio si può notare che si tratta dei passaggi da offerta ad annullamento, quelli che portano ad avere stato attivo e quello da stato attivo a cancellato (essenzialmente su tutti i passaggi dove la transizione avviene per decisione umana, sia essa della azienda o del destinatario). *Modifica di stato*
- Sarà possibile infine salvare l'offerta incompleta (prima che diventi contratto) nel caso non siano disponibili tutti i dati necessari alla sua messa in opera, come nella alternativa 6.a che richiede l'attesa di un sopralluogo da parte di uno degli operatori, o non siano del tutto definitivi, come *Salvataggio dell'Offerta*

nella alternativa 8.a in cui è in corso una contrattazione con il cliente e i dati sono soggetti a variazioni.

Una volta che il ciclo di creazione viene terminato ed il contratto entra in opera è possibile cominciare a programmare gli interventi specificati, ed è quindi possibile spostarsi alla schermata di gestione del calendario ed operare con i meccanismi di schedulazione.

4.5 REALIZZAZIONE DELLA SCHERMATA DI GESTIONE DEL CALENDARIO

Questa sezione del modulo viene pensata per essere in grado di concedere agli utilizzatori del sistema la capacità di poter tenere sotto occhio la distribuzione temporale degli interventi per loro rilevanti e, nel caso degli amministratori, poter riuscire a gestire al meglio le proprie risorse quali operatori, mezzi e attrezzature.

Approccio alla multiutenza

In questo campo troverà applicazione particolare il concetto di multi-utenza definito precedentemente; gli amministratori useranno questa sezione per tenere sotto controllo i piani giornalieri degli interventi, con funzioni di visualizzazione e modifica; gli operatori potranno accedere alla schermata per prelevare i loro piani giornalieri che saranno provvisti di una funzione di stampa; i clienti potranno essere informati sugli interventi che li riguardano e sui dettagli associati.

Abbiamo dunque rilevato le due funzionalità di base che dovranno essere implementate per il corretto funzionamento; la visualizzazione, disponibile per ogni tipologia di utente, in base alla quale personalizza il contenuto che viene visualizzato (ogni visualizza le informazioni che gli sono pertinenti escluso l'amministratore, che è in grado di visualizzare il piano nella sua completezza), e la modifica, disponibile solamente per l'amministratore che porta a formare il piano definitivo degli interventi. In questo paragrafo sarà trattata la visualizzazione, mentre nel successivo sarà trattata nel dettaglio la modifica con la programmazione degli interventi.

Modalità di visualizzazione mensile

Si decide di strutturare la sezione con due modalità di visualizzazione: una su base mensile e una su base giornaliera. La visualizzazione mensile consente di avere un lungo periodo temporale sotto controllo, in modo da avere una visione più generale degli interventi schedulati. Ogni giorno può essere suddiviso ulteriormente nelle sue ore di lavoro per ognuna delle quali è visibile la lista degli interventi che vi sono programmati. I dettagli principali di ogni singolo intervento saranno inoltre accessibili semplicemente scorrendo il puntatore di esse, grazie ad una funzionalità javascript, che ne farà comparire in una nota voci quali contratto di appartenenza, infestante e tipologia di intervento alla quale si dedica, luogo in cui va ad operare. Tuttavia ci sono informazioni che per la loro natura complessa non sono visualizzabili in questo modo, ad esempio la lista di tutti gli operatori e i mezzi assegnati all'intervento. Per risolvere questo inconveniente, ed anche per necessità successiva per le operazioni di modifica, si decide anche di aggiungere la comparsa di un pop-up contenente le informazioni relative all'intervento nella loro completezza, attivato con un click del mouse. Infine, per consentire all'amministratore una migliore visione generale dell'occupazione temporale degli operatori nel piano programmato, si decide di aggiungere all'interno

di ogni cella oraria un indicatore della loro occupazione, dotato anche di un codice di colori variante dal rosso al verde per consentire di trovare meglio a colpo d'occhio gli spazi temporali disponibili per la programmazione degli interventi.

Per consentire di gestire meglio la problematica della gestione degli operatori e del loro piano giornaliero viene inoltre introdotta la modalità di visualizzazione su base giornaliera. Questa modalità sarà accessibile a partire da quella su base mensile semplicemente cliccando sul giorno per il quale si intende visualizzare il piano giornaliero. La visualizzazione viene strutturata in forma tabulare, con gli operatori disponibili posti sulle colonne e le ore di lavoro poste sulle righe. Gli interventi vengono quindi collocati sulle intersezioni tra l'ora in cui sono programmati e gli operatori che vi sono assegnati all'esecuzione. E' possibile così avere una visione dettagliata del piano giornaliero di ciascun operatore, per ognuno dei quali è consentita la stampa come precedentemente anticipato. Sono ancora presenti anche in questa modalità le funzioni di visualizzazione ridotta e completa dei dettagli dei singoli interventi.

Modalità di visualizzazione giornaliera

La schermata di Gestione del calendario funge da piattaforma operativa per le operazioni di schedulazione automatica e schedulazione manuale, per la quale presenta funzioni avanzate che saranno discusse nei paragrafi successivi.

4.6 REALIZZAZIONE DELLA PROGRAMMAZIONE DEGLI INTERVENTI

In questo paragrafo vengono trattate le due modalità di programmazione degli interventi disponibili dalla schermata di gestione del calendario: la schedulazione automatica e la schedulazione manuale. Prima di cominciare a trattarle tuttavia è necessario aprire una parentesi riguardante l'attività di programmazione e le istanze di intervento, in particolare il parametro stato definito durante la progettazione della base di dati. Per come è definita (e successivamente, per come sarà attuata) l'operazione di programmazione degli interventi è una operazione che può richiedere tempo, necessario spesso per contattare il destinatario e mettersi d'accordo con lui, e l'effettuazione di modifiche continue al piano di interventi, necessarie per arrivare ad una pianificazione più efficiente possibile. Per questo motivo ad ogni istanza di intervento programmata o programmabile memorizzata all'interno della base di dati vengono associati tre possibili stati:

Stati delle Istanze di intervento

SCHEDULAZIONE (S) Queste istanze solitamente non sono ancora state ultimate nei loro dettagli, ad esempio può mancare una data e ora per la loro eventuale esecuzione o può non esserci alcun operatore assegnato. Rappresenta uno stato volatile in cui le istanze sono ancora in via di definizione e non sono in alcun modo vincolanti per il piano di schedulazione finale. La decisione di mantenere il salvataggio nella base di dati di queste istanze concede la possibilità di poter continuare una progettazione del piano in più di una sola sessione.

PRENOTAZIONE (P) Rappresenta lo stato in cui si ha una proposta concreta per l'effettuazione dell'intervento, che però deve essere ancora confermato per l'esecuzione; per far questo sarà solitamente necessario veri-

ficare che il destinatario sia effettivamente d'accordo con la scelta effettuata. Non è ancora uno stato vincolante per il piano finale, ma al contrario della schedulazione o non è uno stato volatile, poiché richiede che la data e l'ora, gli operatori, i mezzi e le attrezzature disponibili siano effettivamente disponibili contemporaneamente e vincola la definizione di altre istanze con le stesse caratteristiche.

CONFERMA (c) Definisce lo stato finale di un istanza; è stata ricevuta la conferma per l'esecuzione dell'intervento che quindi diventa definitivamente parte del piano di schedulazione.

4.6.1 Schedulazione automatica degli interventi

La funzionalità di schedulazione automatica degli interventi si occupa di ridurre il tempo necessario a formare il piano definitivo degli interventi; infatti l'output di questa procedura sarà un insieme di istanze definite e che nel complesso raggiunge il miglior compromesso tra gli obiettivi primari che sono richiesti al piano finale, che in ordine di priorità sono:

1. Rispetto dei vincoli impostati fino a questo momento
2. Raggruppamento degli interventi preferenziale in base alla zona in cui sono locati
3. Ottimizzazione degli spostamenti
4. Minor numero possibile di contratti che non riescono a concludere i loro interventi stabiliti prima della loro scadenza

PHP vs Javascript

Una scelta importante che bisogna innanzitutto effettuare per arrivare alla realizzazione di questa funzione riguarda il linguaggio di programmazione che andrà ad implementarla; tra quelli elencati nel capitolo 3 ovvero il dilemma si ha tra l'uso di PHP e quello di Javascript. La scelta in questo caso ricade su PHP; i motivi che portano a questo consistono soprattutto nella natura lato server del linguaggio; l'esecuzione del codice (al contrario di Javascript) non appesantisce in alcun modo il client ed è da esso separata; inoltre per come è stato pensato l'accesso alla base di dati, tramite la libreria Doctrine ORM, l'uso di PHP risulta in ogni caso vincolante per le operazioni necessarie alla programmazione. Per ogni contratto saranno considerate le singole

Sessioni concorrenti

sessioni senza legami vincolanti tra di loro nella fase di schedulazione delle istanze di intervento; la loro caratteristiche variabili anche all'interno dello stesso contratto (ad esempio la possibilità di avere sessioni che operano in luoghi differenti o in periodi diversi) impone che vengano trattate come entità a se stanti. La programmazione delle sessioni richiede anche la decisione tra una schedulazione in grado di stabilire subito tutte le loro istanze o che alternativamente li stabilisce uno alla volta, aspettando il successo nella loro esecuzione prima di programmare il successivo. La scelta in questo caso ricade sulla seconda; in un contesto così incerto e costantemente in cambiamento, in cui può succedere facilmente l'apparizione di future problematiche che possono portare all'annullamento dell'intervento o l'apparizione di interventi più urgenti che ne prendono il posto è possibile avere proposte affidabili solo a breve termine. Le istanze di intervento generate dalla schedulazione sono definite unicamente come istanze di tipo schedulazione o. Questa scelta

Schedulazione intera vs parziale

Istanze di tipo S

porta inevitabilmente alla necessità di una successiva conferma. La decisione viene adottata per dare modo all'utente di verificare che le scelte operate in modalità automatica coincidano con le sue aspettative o di aggiungerci successivamente le proprie modifiche, anche se così facendo si rinuncia a una totale automazione del sistema.

Si decide di dare una struttura ciclica all'algoritmo di schedulazione; ogni ciclo stenderà il piano di interventi relativo ad una giornata lavorativa; ciò è dovuto al fatto che le uscite giornaliere delle squadre di intervento terminano comunque al punto di partenza (l'azienda), il che indica che per ogni giornata l'algoritmo risolutivo del TSP andrà a formare il cosiddetto circuito hamiltoniano (definito a pagina 7) slegato da quello della giornata precedente (ogni giornata finisce sempre con il ritorno al punto di partenza); oltre a questo è necessario valutare per il rispetto dei vincoli diversi parametri che variano su base giornaliera (accessibilità dei luoghi, inizio delle sessioni, intervallo di attesa tra un'istanza e la successiva, ecc.) che richiede ogni volta la loro rivalutazione. Si suppone inizialmente un periodo mensile su cui verrà eseguita la programmazione (quindi 30 cicli giornalieri), ma sarà prevista in futuro la possibilità di variare questo parametro a discrezione dell'utente.

Ciclo giornaliero

Per ogni ciclo di schedulazione vengono individuate le seguenti fasi, successivamente dettagliate:

- Preparazione alla schedulazione
- Formazione dell'insieme di schedulazione giornaliero
- Formazione delle squadre di intervento (Team)
- Riempimento della Matrice di Schedulazione
- Soluzione del problema di ricerca del percorso a costo minimo

Preparazione alla schedulazione

Prima di mettere in atto le operazioni di schedulazione è necessario effettuare il recupero dei dati necessari nelle fasi successive; Il caricamento dei dati da database viene effettuato in blocco all'inizio del ciclo, php provvede successivamente a effettuare l'ordinamento e la strutturazione dei dati ad oggetti; questa scelta è preferita a quella del caricamento da database per ogni singolo elemento per la velocità di esecuzione di php rispetto a quella delle operazioni di accesso al database; il tempo impiegato da php per riordinare una serie di dati disorganizzati risulta minore rispetto al caricamento organizzato, ma effettuato in più occasioni (quindi con più accessi al database), dei dati tramite query.

I dati che sono richiesti per operare nelle fasi successive sono i seguenti:

- Le sessioni che sono effettivamente eseguibili in giornata; tra tutte le sessioni esistenti nella base di dati si considerano solo quelle che soddisfano le seguenti caratteristiche: sono appartenenti a contratti in stati attivi e che non hanno ancora raggiunto la loro scadenza, non hanno ancora completato i loro interventi, non hanno istanze già prenotate o schedulate nei cicli precedenti, presentano un intervallo di giorni sufficiente dall'ultima istanza confermata ed eseguita con successo; le tipologie di intervento su cui si va ad agire;
- i luoghi di destinazione;

- le eventuali zone di cui essi eventualmente fanno parte;
- I vincoli di accessibilità per i luoghi su cui vanno ad agire;
- I vincoli post-intervento che devono essere considerati nella fase di scelta degli operatori, mezzi e attrezzature assegnate;
- Le istanze P e C esistenti precedentemente alla programmazione; la schedulazione automatica è stata pensata per adeguarsi alle scelte precedentemente effettuate dagli utilizzatori, e per questo le nuove istanze terranno conto degli istanti temporali già impegnati e delle risorse già assegnate;
- Le risorse disponibili per la schedulazione, ovvero operatori, mezzi e attrezzatura;

Il codice seguente rappresenta una Query MySQL d'esempio per il recupero dei dati relativi alle sessioni attive.

```

SELECT Sessione.*, Contratto.codice, Contratto.data_scadenza, Luogo
.indirizzo, Luogo.n_civico, Luogo.zona, Luogo.latitudine, Luogo
.longitudine, Luogo.comune, TI.nome_tipo_intervento, TI.
id_infestante, TI.nome_infestante
FROM Sessione
INNER JOIN (
  SELECT Contratti, Contratto.codice, Contratto.data_scadenza
  FROM Contratto
  WHERE Contratto.tipo_contratto = 'OC' OR Contratto.
  tipo_contratto = 'Spot'
) AS Contratto ON Sessione.contratto = Contratto.id
LEFT JOIN (
  SELECT Luogo.id_luogo, Luogo.indirizzo, Luogo.n_civico,
  Luogo.latitudine, Luogo.longitudine, Luogo.zona,
  Comune.id, Comune.nome
  FROM Luogo
  LEFT JOIN Comune ON Luogo.comune = Comune.id
) AS Luogo ON Sessione.luogo = Luogo.id
LEFT JOIN (
  SELECT TipoIntervento.id, TipoIntervento.nome as
  nome_tipo_intervento, Infestante.id as id_infestante,
  Infestante.nome as nome_infestante
  FROM TipoIntervento
  LEFT JOIN Infestante ON TipoIntervento.infestante = Infestante.
  id
) AS TI ON Sessione.tipo_intervento = TI.id
LEFT JOIN (
  SELECT Istanza.sessione, COUNT(*) as i_p
  FROM Istanza
  WHERE Istanza.stato = 's' OR (Istanze.stato = 'p' AND Istanze.
  data >= '<data_di_schedulazione>')
  GROUP BY Istanza.sessione
) AS Istanze ON Sessione.id = Istanze.sessione
LEFT JOIN (
  SELECT Istanza.sessione, MAX(Istanze.data) as last

```

```

FROM Istanza
WHERE Istanza.stato = 'c' AND Istanza.successo = true
GROUP BY Istanza.sessione
) AS IstanzeConferma ON Sessione.id = IstanzeConferma.sessione
WHERE Sessione.interventi_effettuati < Sessione.n_interventi
AND Sessione.inizio_sessione <= '<data_di_schedulazione>'
AND (Istanze.i_p is null OR Istanze.i_p <= 0)
AND (IstanzeConferma.last is null OR DATEDIFF('<
data_di_schedulazione>', IstanzeConferma.last) >= Sessione.
intervallo_interventi)

```

Formazione dell'insieme di schedulazione giornaliero

La problematica successiva che viene gestita è la scelta delle sessioni, tra quelle disponibili caricate nella fase precedente, per le quali sarà effettivamente schedulato un intervento in giornata. Il motivo che ci spinge a effettuare questa riduzione è banale: una giornata dispone di ore di lavoro e risorse finite, motivo per cui in casi di elevati numeri di sessioni attive è fisicamente impossibile la loro esecuzione totale in una unica giornata; Il primo passo per compiere una decisione del genere è quindi trovare un upper bound, ovvero il limite massimo di interventi schedulabili in una singola giornata. Per arrivare a questa informazione tuttavia è necessario effettuare una introduzione di alcune entità che vengono introdotte a livello concettuale; lo spazio disponibile per l'impostazione degli interventi è calcolato sulle dimensioni temporale e delle risorse umane e non. Se supponiamo di assegnare alla dimensione temporale una suddivisione in blocchi della durata di mezz'ora l'uno otteniamo un passaggio da una funzione continua del tempo ad una discreta. È possibile quindi esprimere lo spazio disponibile come una matrice finita le cui dimensioni sono il tempo e le risorse. L'area coperta da questa matrice rappresenta quindi la massima dimensione schedulabile per una giornata lavorativa e quindi per ottenere un processo ottimale dovrà essere considerato almeno un numero di sessioni tale da riuscire a coprire completamente questa area.

Determinazione del limite delle sessioni

A questo punto sorge una ulteriore problematica. Con che criterio dovranno essere scelte tra quelle disponibili le istanze che andranno ad essere schedulate? Viene quindi definito un indice di priorità che viene assegnato a ciascuna sessione e che ne stabilisce l'ordine di preferenza nella scelta. Questo parametro viene calcolato da una funzione che prende come input i seguenti parametri: il tempo necessario per la sessione ad arrivare al completamento (che tiene conto dell'intervallo in giorni che è necessario aspettare tra un intervento ed il successivo ed il numero effettivo di interventi ancora da effettuare); il tempo effettivamente disponibile per terminare la sessione prima della scadenza; la politica adottata prevede che in caso si rilevi una sessione che non sia in grado fisicamente di rispettare la scadenza (ovvero con tempo necessario maggiore del tempo disponibile) la sua priorità risulta nulla, ovvero la sessione non va ad occupare il tempo che può permettere ad altre di rispettare le loro scadenze e quindi si punta a ridurre il numero di sessioni che falliscono il deadline, nel rispetto degli obiettivi di schedulazione prefissati in precedenza; il numero di ore e giorni in cui il luogo assegnato ad una sessione è accessibile durante la settimana; bisogna notare che la sessio-

Il criterio di priorità

ne non sarà considerata nel caso si stia effettuando la schedulazione per un giorno in cui il luogo assegnatogli non è accessibile;

La lista delle sessioni viene quindi ordinata in base a questo indice usando una funzione di ordinamento di tipo quick-sort.

Le Zone e i Workgroup

In questa fase è anche necessario tenere conto delle sessioni che sono schedulate in luoghi appartenenti ad una zona comune. La richiesta del committente era che, per rispettare il suo modo di operare, i luoghi appartenenti ad una stessa zona ricevano possibilmente gli interventi in una stessa giornata, in caso questo non fosse possibile in giornate adiacenti. Il modo di agire a questo riguardo porta a definire quindi l'entità concettuale gruppo di lavoro o *Workgroup*, che consiste dell'insieme delle sessioni attive che si svolgono all'interno di una stessa zona. Nella fase di formazione dell'insieme di schedulazione è innanzitutto necessario stabilire se il *Workgroup* è programmabile in giornata. Questa decisione viene effettuata basandosi su informazioni quali il numero di sessioni appartenenti al *Workgroup* pronte alla schedulazione, il numero totale delle sue sessioni attive e i giorni passati dall'ultima schedulazione giornaliera del *Workgroup*. Si avrà quindi per ogni *Workgroup* un periodo temporale in cui esso non è schedulabile poichè è necessario attendere che altre sessioni prossime ad essere pronte alla schedulazione lo diventino (periodo di attendibilità) ed un altro periodo prioritario, in cui è trascorso una brevissima finestra temporale dall'ultima giornata di esecuzione del *Workgroup* ed è quindi necessario portarlo a conclusione abbastanza rapidamente (periodo continuativo). Successivamente viene definita una priorità di *Workgroup*, analoga a quella delle sessioni singole, che tiene conto delle priorità singole di quelle che ne fanno parte e eventualmente della sua presenza nel periodo continuativo appena definito. In fase di formazione dell'insieme di schedulazione sarà quindi usata la priorità di *Workgroup* per effettuare il suo ordinamento di scelta rispetto alle altre sessioni singole attive ed agli altri *Workgroup*, mentre all'interno del *Workgroup* l'ordinamento delle sessioni continuerà ad avere luogo tramite la priorità standard. E' possibile notare infine che nel caso non siano presenti sessioni legate a zone l'algoritmo di schedulazione funziona in modo indistinto da quello descritto inizialmente.

L'insieme di sessioni scelte in questa fase, unito a quello delle istanze decise precedentemente all'avviamento dell'algoritmo di schedulazione costituisce la base per la formazione delle squadre di intervento, definite nel paragrafo seguente.

Formazione delle squadre di intervento

In questa fase viene definita la seconda dimensione della matrice dello spazio di schedulazione, menzionata precedentemente semplicemente come dimensione delle risorse. Gli elementi che vanno a comporre l'insieme delle risorse disponibili per l'esecuzione degli interventi consistono negli Operatori, le persone che compiono effettivamente l'intervento, i Mezzi, attraverso i quali si compiono gli spostamenti tra i luoghi dove avvengono gli interventi e le Attrezzature, che possono essere usate dagli operatori durante gli interventi. Viene definita a questo punto una entità a livello concettuale che andrà a formare la seconda dimensione della matrice di schedulazione, ovvero l'entità squadra di intervento o *Team*. Ogni *Team* sarà composto dall'insieme di uno o più operatori, uno o più mezzi e le attrezzature caricate su di essi.

I Team

Formazione dei Team

La formazione dei *Team* verrà effettuata tenendo conto dell'insieme di ses-

sioni formato precedentemente; viene definito un algoritmo di definizione dei Team che agisce ciclicamente, finché non è più possibile creare altri Team o finché non sono terminate le sessioni da esaminare. Le sessioni vengono valutate ad ogni ciclo seguendo l'ordine di priorità definito su di esse; vengono valutati i Team creati nei cicli precedenti per verificare la loro adeguatezza ad operare un intervento per la sessione in esame; viene per questo controllato che gli operatori siano abilitati ad effettuare la tipologia di intervento, che sia presente l'attrezzatura necessaria e che siano rispettati tutti i vincoli post-intervento definiti. Nel caso venga riscontrata l'adeguatezza di almeno un Team, questo viene aggiunto alla lista dei Team adeguati all'esecuzione dell'intervento considerato ed inoltre si aumenta il conto delle istanze abilitate all'esecuzione da tale Team. Inoltre si tiene conto del tempo disponibile nella giornata lavorativa che ogni Team potenzialmente andrà ad occupare. In caso non si abbia alcun riscontro positivo o non si trovino Team adeguati e con tempo disponibile all'esecuzione invece si cerca di creare un Team fatto su misura per l'esecuzione della sessione esame, che viene assegnato ad essa.

Riempimento della Matrice di Schedulazione

Una volta definiti i Team specifici che opereranno in questa giornata è possibile procedere con la definizione della *Matrice di schedulazione*. Come menzionato precedentemente, la matrice opera sulle due dimensioni del tempo e dei Team definiti; è quindi visionabile come una tabella che presenta sulle righe il tempo, diviso in intervalli di mezzora, e sulle colonne i Team. Il compito della matrice è memorizzare gli spazi che sono già impegnati e trovarne di liberi per le istanze di intervento che si vanno a schedulare. Innanzitutto quindi si procede ad impegnare gli spazi di occupazione certa, ovvero quelli che sono già riservati per le istanze di intervento prenotate o confermate precedentemente all'avviamento dell'algoritmo. Successivamente si valutano ciclicamente, sempre in ordine di priorità, le sessioni da schedulare; la classe Matrice cercherà quindi di assegnare un posto per schedulare l'istanza. La scelta del Team ricade potenzialmente sul Team, tra quelli adeguati a svolgere l'intervento, che ha presentato una compatibilità minore durante la fase precedente; questa scelta mira a fare in modo che i Team maggiormente compatibili restino disponibili il più a lungo possibile, per evitare situazioni in cui risulta impossibile collocare un'istanza sulla matrice per mancanza di spazio in corrispondenza dei Team adeguati. La scelta dell'orario viene invece valutata sull'orario di lavoro giornaliero, definito precedentemente, che tiene conto anche delle pause lavorative e dei posti riservati inizialmente alle istanze preesistenti o prenotati nei cicli precedenti. Si procede così a trovare una collocazione temporale e un Team adeguato per ogni intervento finché questi non sono esauriti.

Soluzione del problema di ricerca del percorso a costo minimo

L'ultima fase dell'algoritmo consiste nell'ottimizzazione dei piani giornalieri dei Team definiti nella fase precedente; come anticipato tale risoluzione sarà trattata in modo analogo a quella del problema np-completo del commesso viaggiatore (TSP).

La risoluzione assume una forma ciclica; infatti per il singolo giorno non viene trattato un singolo percorso, bensì uno per ogni Team che è stato formato. Il problema che ci poniamo nella pratica è quindi quello di trovare

il percorso minimo che collega le destinazioni degli interventi di disinfestazione. L'algoritmo quindi valuta ciclicamente le colonne della matrice precedente, che rappresentano il piano giornaliero di ogni Team (anche riferito successivamente come uscita o escursione).

Concettualizzazione del problema

La prima operazione che deve essere effettuata consiste nel passaggio dal problema fisico a quello concettuale; di conseguenza i luoghi visitati dall'escursione diventano i vertici del grafo del TSP, mentre per i suoi archi è considerato il tragitto che li collega. Il peso degli archi è trattato come lunghezza spaziale di questo tragitto; a partire dalle coordinate di dei due luoghi di cui si vuole calcolare (due coppie consistenti in latitudine e longitudine) è facilmente derivabile tramite l'uso del teorema di Pitagora. Per la possibilità di posizionare effettivamente su un piano i vertici del grafo (i luoghi) e trovarne così tutte le distanze è possibile affermare che tali distanze sono euclidee.

Determinazione dell'algoritmo risolutivo

Successivamente bisogna determinare l'algoritmo da utilizzare che effettivamente ci permetterà di risolvere il problema. La soluzione più diretta consisterebbe in un metodo di forza bruta, ovvero il tentativo di tutte le possibili permutazioni degli interventi facenti parte di una escursione. Questo metodo garantisce di trovare la soluzione migliore al problema; tuttavia questo porta ad avere complessità computazionali piuttosto elevate, situate su una scala di n fattoriale, con n pari al numero di istanze facenti parte dell'escursione; l'applicazione di questo metodo risulta quindi sconsigliata. L'approccio che si sceglie di attuare è invece uno basato sulla programmazione lineare, analogo a quello precedentemente descritto durante l'analisi teorica del problema. Si stima che per ogni uscita considerata si abbia a che fare con un problema che tratti un numero massimo di una ventina di locazioni. Un metodo in grado di gestire in modo abbastanza efficace un simile contesto è quello di Branch and Bound.

Branch and Bound

Gli algoritmi di tipo *Branch and Bound*, anche detti di enumerazione implicita, sono algoritmi che applicano un procedimento di tipo *Divide and Conquer*, ovvero che riconduce la risoluzione di un problema difficile di programmazione lineare intera alla risoluzione di 2 sottoproblemi, anch'essi di PLI ma più semplici. La descrizione formale del metodo di applicazione dell'algoritmo è la seguente: Sia $\min c^T x : Ax = b, x \geq 0$ intero il problema di PLI iniziale (come ad esempio quello di pagina 9 relativo al Traveling Salesman Problem). Si risolve il rilassamento continuo $\min c^T x : Ax = b, x \geq 0$, individuando una soluzione base ottima x^* . Se x^* è una soluzione intera (come indica il problema di PLI di pagina 9 le variabili di decisione della soluzione possono assumere solo valori pari ad 1 o 0, che equivalgono alla scelta o meno dell'arco in questione) non è necessario procedere ulteriormente. Altrimenti, si sceglie una variabile x_{*h} frazionaria e si costruiscono i due sottoproblemi:

$$PLI_2 : \min\{c^T x : Ax = b, x_h \leq \lfloor x_{*h} \rfloor, x \geq 0 \text{ intero}\}$$

$$PLI_3 : \min\{c^T x : Ax = b, x_h \geq \lceil x_{*h} \rceil, x \geq 0 \text{ intero}\}$$

Questa operazione viene detta di branching; la variabile x_h viene quindi detta variabile di branching e i problemi PLI_2 e PLI_3 si dicono figli di PLI_1 , ottenuti da lui aggiungendo un vincolo di branching. E' possibile quindi risolvere PLI_2 e PLI_3 , in sequenza, riapplicando ricorsivamente il procedimento: per ciascuno dei due sottoproblemi si risolve il rilassamento continuo e se necessario si effettua ancora l'operazione di branching. Si ottiene così una successione gerarchica di sottoproblemi sempre più vincolati e quindi

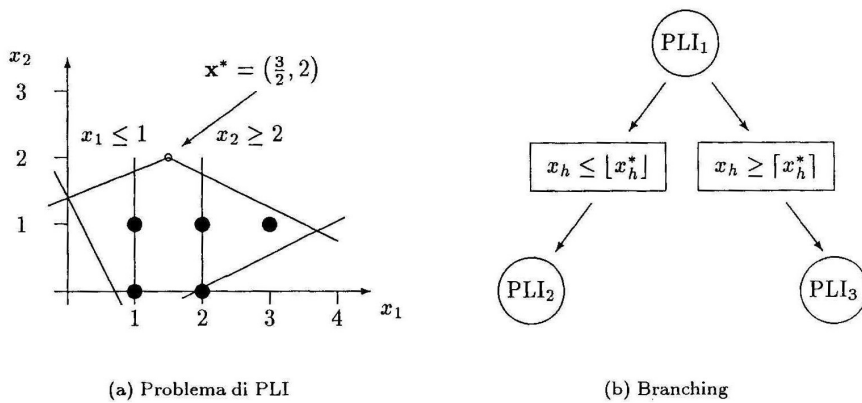


Figura 17: Operazione di branching per un problema di PLI

più facilmente risolvibili. Per descrivere l'esecuzione è possibile usare una notazione come quella di figura 18, detta *albero decisionale*.

L'ottimalità del metodo di Branch and Bound deriva dalla non-obbligatorietà a considerare esplicitamente tutti i nodi presenti nell'albero decisionale. Esistono infatti tre criteri, detti di fathoming, che ci considerano di considerare come sondato un nodo t e quindi di non avere il bisogno di considerarne i figli. I tre criteri sono i seguenti:

Criteri di fathoming

1. la soluzione $x^*_{PL_t}$ del rilassamento continuo PL_t è intera;
2. il rilassamento continuo PL_t è impossibile;
3. $LB_t \geq z_{OPT}$, dove $LB_t := c^T x^*_{PL_t}$ e z_{OPT} è il costo della soluzione ottima intera corrente (criterio di bounding)

Una realizzazione dell'algoritmo di Branch and Bound deve comunque stabilire una regola con cui selezionare il nodo da elaborare nella successiva iterazione; le due tecniche principali sono:

DEPTH FIRST consiste nella scelta sistematica del nodo non ancora elaborato più profondo, simulando un procedimento di tipo ricorsivo: dato un nodo padre si considera immediatamente il primo dei suoi due figli, finché una delle condizioni di fathoming non costringe l'algoritmo a risalire di livello;

BEST-BOUND FIRST viene scelto per l'elaborazione il nodo non elaborato con LB_t minimo, e quindi verosimilmente alla soluzione ottima.

Nel caso pratico in esame la scelta ricade sul Best-bound: ciò comporta una minore generazione di nodi, ma porta l'elaborazione a rimanere a livelli poco profondi dell'albero decisionale, e quindi ad i problemi sono poco vincolati e difficilmente portano ad un aggiornamento della soluzione ottima corrente. Per ridimensionare questo svantaggio si cercherà di arrivare ad individuare in modo euristico una buona soluzione intera; si cercherà quindi di fermare il branching quando la differenza fra la soluzione trovata e il lower bound diventa inferiore rispetto ad una certa soglia. Questo metodo serve a stabilire

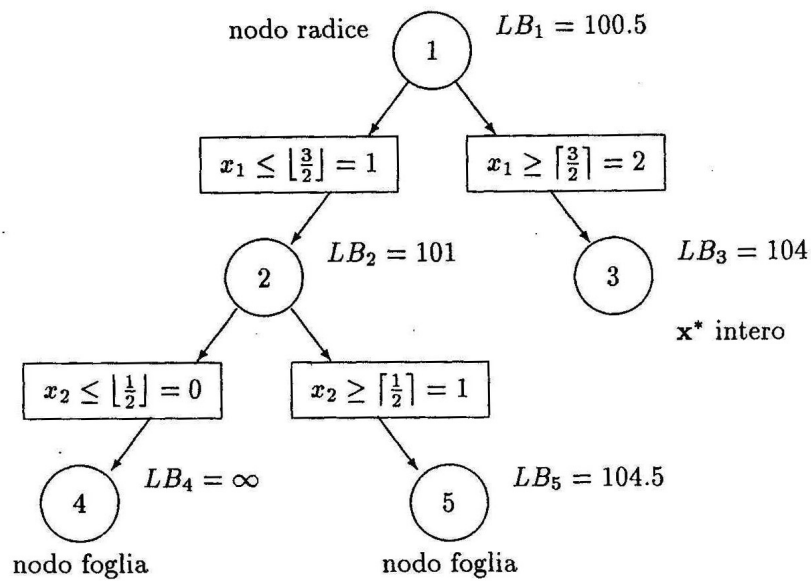


Figura 18: Albero decisionale o Branching Tree

che la soluzione trovata è abbastanza buona per i nostri scopi e si avvicina con un margine accettabile a quella migliore. Il vantaggio ottenuto è una notevole riduzione dei tempo di calcolo.

Considerazione sui vincoli

Spesso purtroppo non è possibile applicare alcune delle operazioni di ottimizzazione del percorso; questo succede nei casi in cui si è in presenza di vincoli di accessibilità per alcune delle istanze di intervento o quando alcuni degli interventi dell'escursione erano esistenti precedentemente alla schedulazione, ovvero in stato di prenotazione o conferma; l'algoritmo di risoluzione descritto si adatterà quindi a gestire anche tale evenienza. In questo ultimo caso è possibile ridurre il problema permettendo innanzitutto all'algoritmo di applicare l'ottimizzazione del percorso prima di procedere ad effettuare le prenotazioni e conferme manuali. Tuttavia non è possibile fare nulla per evitare il problema dei limiti all'accessibilità dei luoghi; quindi non ci si può aspettare una ottimizzazione completa dei percorsi nei casi pratici in cui siano presenti molti vincoli a cui gli interventi si devono attenere.

Pseudocodice relativo all'algoritmo di schedulazione automatica

```

begin
  for(giorno=0; giorno<n; giorno++) //n inteso come numero di
    giorni da schedulare
  begin
    spazio_riservato:=0;

    /* Preparazione alla schedulazione */

    accedo al database;
    prelevo gli operatori, mezzi e attrezzature disponibili
  
```

```

Algoritmo Branch-and-Bound ;
begin
   $m := 1$ ;  $padre[1] := 0$ ;  $Q := \emptyset$ ;
   $z_{OPT} :=$  valore soluzione euristica (eventualmente  $+\infty$ );
  risolvi il rilassamento continuo  $\min\{c^T x : Ax = b, x \geq 0\}$ , e sia
   $x^*$  la soluzione ottima individuata;
   $LB[1] := c^T x^*$  ;
  if ( $x^*$  intera) and ( $c^T x^* < z_{OPT}$ ) then
    begin
       $x_{OPT} = x^*$ ;  $z_{OPT} := c^T x^*$ 
    end ;
  if  $LB[1] < z_{OPT}$  then
    begin
      scegli la variabile frazionaria  $x_h^*$  di branching;
       $vbranch[1] := h$ ;  $valore[1] := x_h^*$ ;
       $Q := \{1\}$ 
    end ;
  while  $Q \neq \emptyset$  do /* elabora i nodi foglia attivi*/
    begin
      scegli un nodo  $t \in Q$ , e poni  $Q := Q \setminus \{t\}$  ;
       $h := vbranch[t]$ ;  $val := valore[t]$  ;
      for  $figlio := 1$  to 2 do /* genera i figli del nodo  $t$  */
        begin
           $m := m + 1$  ;
          if  $figlio = 1$ 
            then  $padre[m] := t$ 
            else  $padre[m] := -t$  ;
          definisci il problema  $PL_m$  associato al nodo  $m$ 
            (vincoli di  $PL_t$  più  $x_h \leq [val]$  se  $figlio = 1$ ,
            oppure  $x_h \geq [val]$  se  $figlio = 2$ ) ;
          risolvi il problema  $PL_m$ , e sia  $x^*$  la soluzione ottima trovata ;
           $LB[m] := c^T x^*$ ;
          if ( $x^*$  intera) and ( $c^T x^* < z_{OPT}$ ) then
            begin /* aggiorna la soluzione ottima */
               $x_{OPT} := x^*$ ;  $z_{OPT} := c^T x^*$  ;
               $Q := Q \setminus \{j \in Q : LB[j] \geq z_{OPT}\}$ 
            end ;
          if  $LB[m] < z_{OPT}$  then
            begin
              scegli la variabile frazionaria  $x_k^*$  di branching ;
               $vbranch[m] := k$ ;  $valore[m] := x_k^*$  ;
               $Q := Q \cup \{m\}$ 
            end
          end
        end
      end
    end
  end .

```

Figura 19: Pseudocodice relativo all'algoritmo di Branch and Bound

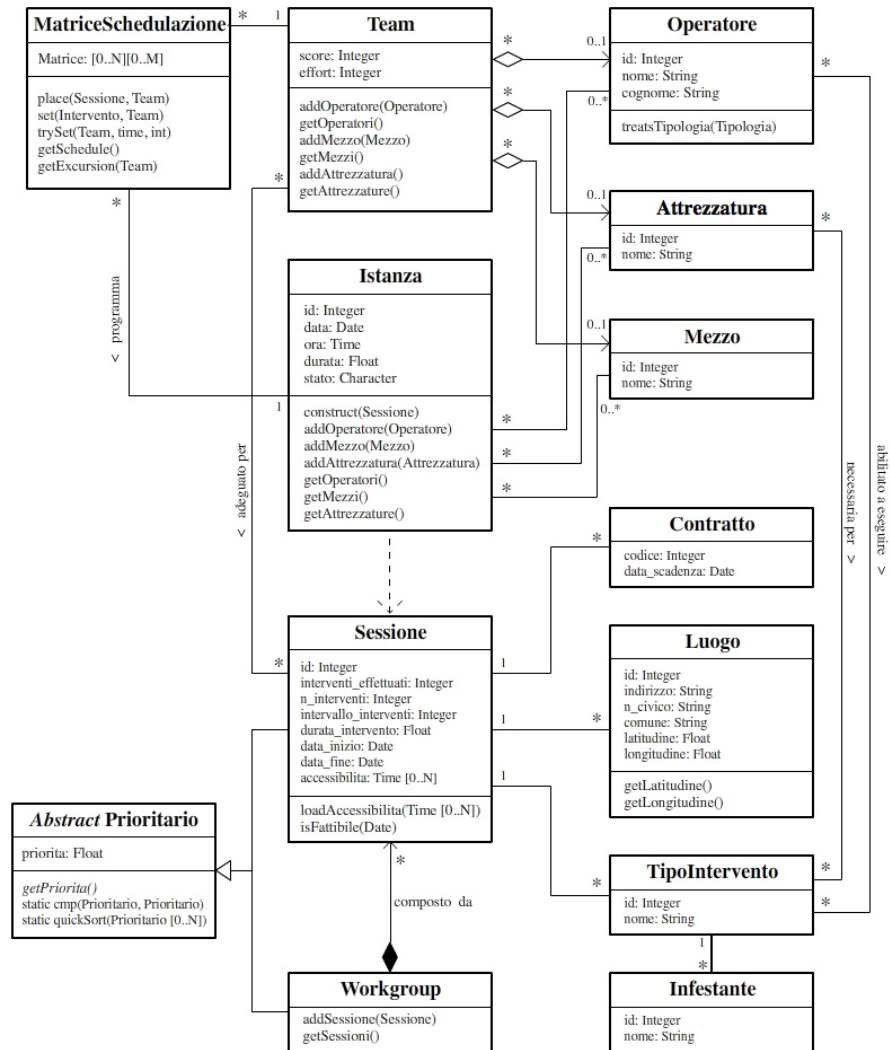


Figura 20: Diagramma delle classi relativo all’algoritmo di schedulazione automatica

```

    per la schedulazione;
prelevo le istanze preesistenti per la giornata e i loro
    dati collegati;
prelevo le sessioni attive per il giorno considerato;
prelevo i vincoli;

for each(Istanza)
begin
    ordino i dati prelevati per la programmazione object-
        oriented;
    spazio_riservato:=spazio_riservato+Istanza.durata;
    aggiungo alla lista delle istanze preesistenti;
end

for each(Sessione)
begin
    ordino i dati prelevati per la programmazione object-
        oriented;
    se in grado di terminare prima della scadenza
    begin
        calcolo il suo indice di priorit ;
        se la sessione fa parte di una zona
            for each(Workgroup in ListaWorkgroup)
            begin
                se la Sessione fa parte della zona assegnata
                    al Workgroup
                    aggiungi la Sessione al Workgroup
            end
            se non stato trovato un Workgroup esistente
                per la zona
                crea nuovo Workgroup, assegnagli la Sessione e
                aggiungilo alla lista
            aggiungo alla lista degli elementi prioritari da
                schedulare
        end
    end
end

/* Formazione dell'insieme di schedulazione giornaliero */

ordino la lista degli oggetti prioritari con la funzione
    quickSort per priorit decrescente;

stimo come max_team il numero massimo di Team che potr
    essere creato nella fase successiva
spazio_disponibile:= max_team*spazio_giornaliero
spazio_riservato;

spazio_richiesto:=0;
i:=0
while ( spazio_richiesto<spazio_disponibile and i <
    ListaPrioritari.length )
begin

```

```

PrioritarioCorrente = ListaPrioritari[i];
se spazio_richiesto + PrioritarioCorrente.durata <=
  spazio_disponibile
begin
  spazio_richiesto:=spazio_richiesto +
    PrioritarioCorrente.durata;
  aggiungi PrioritarioCorrente alla lista degli
    schedulabili;
end
i++;
end

/* Formazione delle squadre di intervento */

for each(Istanza schedulata)
begin
  for(i=0; i<ListaTeam.length; i++)
  begin
    TeamCorrente = ListaTeam[i];
    se TeamCorrente compatibile con l'Istanza
    begin
      assegna il Team all'Istanza;
      TeamCorrente.effort:=TeamCorrente.effort+Istanza.
        durata;
      break;
    end
  end
  se non ho trovato un team compatibile
  begin
    crea Team formato da Operatori, Mezzi e Attrezzature
      richieste dall'Istanza
    se riesco a formare il Team
    begin
      assegna il Team all'Istanza;
      Team.effort:=Team.effort+Istanza.durata;
      aggiungi il Team alla lista dei Team;
    end
    altrimenti
      // situazione in cui alcuni vincoli siano stati
        ignorati nella schedulazione manuale
  end
end

for each (Sessione schedulabile)
begin
  trovato:=false
  for(i=0; i<ListaTeam.length; i++)
  begin
    TeamCorrente = ListaTeam[i];
    se TeamCorrente adeguato per la sessione
    begin
      aggiungi il Team come compatibile alla sessione;
    end
  end
end

```

```

        TeamCorrente.score++;
        se TeamCorrente ha ancora tempo non occupato e
            trovato=false
        begin
            trovato:=true;
            TeamCorrente.effort:=TeamCorrente.effort+
                Sessione.durata;
        end
    end
end
se non ho trovato un team compatibile
begin
    crea Team formato da Operatori, Mezzi e Attrezzature
        adeguati per la sessione e i suoi vincoli
    se riesco a formare il Team
    begin
        aggiungi il Team come compatibile alla Sessione;
        Team.effort:=Team.effort+Sessione.durata;
        aggiungi il Team alla lista dei Team;
    end
    altrimenti
        // la sessione non sar schedulabile in questa
            giornata
    end
end

/* Riempimento della matrice di schedulazione */

creo l'oggetto MatriceSchedulazione, dotato di tante
    colonne quanti sono i Team creati

for each (Istanza schedulata)
begin
    se Istanza ha un Team compatibile
        for(i=0; i<Istanza.durata; i++)
        begin
            impegna MatriceSchedulazione[Team][Istanza.ora+i
                ];
        end
    end
end

for each (Sessione schedulabile)
begin
    ordina i Team assegnati alla Sessione in ordine di
        punteggio crescente
    for each(Team in Sessione.ListaTeam)
    begin
        se Team adeguato a svolgere la Sessione
        begin
            crea un oggetto Istanza a partire dai dati della
                Sessione
            assegna Operatori, Mezzi e Attrezzatura del Team

```

```

        trovato alla nuova Istanza
        aggiungi l'Istanza alla lista delle nuove Istanze
        da aggiungere al database
        for(i=0; i<Istanza.durata; i++)
        begin
            impegna MatriceSchedulazione[Team][Istanza.ora
                +i];
        end
        break;
    end
end
end

/* Soluzione del problema di ricerca del percorso a costo
    minimo */

for each(Team)
begin
    ottieni dalla MatriceSchedulazione l'uscita e le
        Istanze relativi al Team
    risolvi il problema di programmazione lineare intera
        relativo al TSP
    modifica Istanza.ora per le Istanze il cui piano viene
        cambiato
end

/* Salvataggio nella base di dati */

for each(Istanza in Lista nuove Istanze)
begin
    salva nel database l'Istanza
    salva nel database le relazioni con Operatori, Mezzi e
        Attrezzature collegate
end
end
end
end

```

4.6.2 Schedulazione manuale

La modalità di schedulazione manuale va a complementare le funzionalità di quella automatica; tramite una serie di controlli intuitivi viene implementata sulla schermata di gestione del calendario per permettere all'amministratore del sistema di applicare sul piano di schedulazione operazioni come la creazione, la modifica e la conferma delle istanze di intervento.

Barra di schedulazione

Se si intende cominciare le operazioni di schedulazione con la modalità manuale piuttosto che quella automatica, è prevista una funzionalità aggiunta della schermata giornaliera del calendario, denominata *barra di schedulazione*, in grado di suggerire le istanze di intervento appartenenti alle sessioni attive della giornata che sono disponibili per la creazione.

Le operazioni di modifica

Le operazioni di modifica manuali dell'intervento consistono nel cambio di data, ora, durata e assegnazione di operatori, mezzi, e attrezzature al singolo

intervento. Queste operazioni sono applicabili in due modi distinti: il primo è da popup, attivabile con un click del mouse sia dalla schermata mensile che da quella giornaliera; qui viene presentato un modulo dettagliato con tutte le informazioni relative all'intervento selezionato; le informazioni strettamente pertinenti all'istanza di intervento sono modificabili e presentano, analogamente a quanto avveniva per la gestione delle anagrafiche, un sistema di controllo dei campi, che in caso di errori di compilazione evidenzia i campi da correggere prima di poterne effettuare l'invio, che viene eseguito anche in questo caso tramite procedura ajax. Il secondo avviene tramite modifica visuale sulla schermata giornaliera; ad alcune operazioni di modifica è facilmente associabile un'operazione intuitiva, che ne permette la riduzione dei tempi di applicazione in modo considerevole. Per questo motivo le seguenti operazioni sono eseguibili in modo alternativo applicando le funzionalità fornite dalla libreria jQuery UI:

MODIFICA DELLA DURATA è sufficiente trascinare il bordo inferiore della cella rappresentante l'intervento al quale si vuole applicare la modifica per ridurne o aumentarne la durata;

CAMBIO DI ORARIO è possibile selezionare e trascinare la cella di un intervento per spostarlo sulla riga di un orario diverso;

AGGIUNTA/CAMBIO DI OPERATORI è possibile selezionare e trascinare la cella di un intervento per spostarlo sulla colonna relativa ad un altro operatore: un messaggio di scelta permetterà di decidere se effettuare il passaggio rimuovendo l'operatore originale oppure mantenendolo.

In ognuno dei casi, le operazioni di modifica devono tener conto dei vincoli definiti; nello specifico si tratta dei vincoli post-intervento definiti sulle sessioni a cui le istanze di intervento si riferiscono, dei vincoli di accessibilità assegnati ai luoghi di destinazione e delle violazioni di assegnazione (che si hanno ad esempio se si cerca di assegnare un intervento ad un operatore mentre questo è già impegnato ad eseguirne un altro). Tuttavia si sceglie in questo contesto un modo diverso di trattare i vincoli rispetto a quanto avviene durante la schedulazione automatica; la violazione di ciascuno comporterà, nel momento in cui si cerca, consapevolmente o meno, di effettuarne la violazione, la comparsa di un alert, ovvero un messaggio di notifica il cui compito è rendere noto all'utente il potenziale errore logico che sta per essere applicato. A questo punto l'utente è in grado di decidere se continuare ugualmente con la modifica che intendeva apporre all'inizio o se annullare l'operazione. La scelta di non rendere inviolabili i vincoli in questa fase serve a permettere all'amministratore di gestire i rari casi in cui, per motivi imprevedibili, l'assegnazione di un intervento richieda un trattamento non previsto dal sistema di schedulazione. E' possibile comunque fare assumere alle istanze un comportamento più canonico definendo su di esse un vincolo generico che descrive il caso in esame e come deve essere trattato. E' necessario infine notare che la presenza di tale meccanismo richiede un certo livello di consapevolezza nell'uso da parte dell'utente incaricato.

L'altra operazione concessa oltre alla modifica è quella del cambio di stato; è quindi possibile in questo modo lavorare in più momenti, creando prima le schedulazioni o e dopo averle definite nei dettagli effettuarne la prenotazione, ed infine la conferma. E' consentito annullare le prenotazioni di un intervento e ritornare in stato di schedulazione o nel caso in cui le contrattazioni con il

Valutazione dei vincoli

Il cambio di stato

cliente collegato non vadano a buon fine. Dopo la conferma e l'avvenimento di un intervento viene inoltre offerta la possibilità di comunicare il successo o insuccesso dell'operazione e di inviare un documento di rapporto che ne descrive il suo svolgimento e le motivazioni.

*Alternanza dei due metodi di
schedulazione*

Lo sviluppo delle due modalità di schedulazione è essenzialmente effettuato in modo da permettere la loro alternanza; le schedulazioni o generate tramite la schedulazione automatica saranno successivamente visibili dalla schermata di gestione del calendario; analogamente dopo aver fatto modifiche e prenotazioni in modalità manuale è possibile lanciare la schedulazione automatica, che effettua la programmazione adeguandosi alle scelte effettuate precedentemente durante quella manuale.

5

INTERFACCIA GESTIONALE WEB

Questo capitolo viene dedicato alla descrizione dell'interfaccia del modulo realizzato, allo scopo anche di servire come una guida all'uso per l'utente finale.

L'accesso al modulo comincia attraverso una *schermata di login*, che come menzionato precedente è condivisa con il sistema IES nel quale esso va ad integrarsi. Questa schermata va compilata con il nome dell'utente che ci è stato assegnato e la sua relativa password. La pressione del tasto invio oppure del tasto login permette quindi l'ingresso vero e proprio al gestionale, avviando la sessione utente; se il sistema dovesse restare inattivo per un periodo superiore ai 5 minuti la sessione scadrebbe, portando quindi la schermata di login ad essere riproposta.

Schermata di Login

L'interno del modulo applicativo trattato possiede una barra dei menù superiore, che sarà usata per navigare tra le funzionalità di IES come ad esempio la gestione degli utenti, ed una barra collocata subito sotto ad essa, in grado di fornire l'accesso ai due menù principali delle anagrafiche e degli interventi, all'interno dei quali sono collocate anche le funzionalità menzionate in precedenza. Il click sinistro del mouse sul menù Anagrafiche fa comparire una tendina, nella quale è possibile scegliere l'anagrafica che si vuole arrivare a gestire.

La schermata che compare accedendo ad una sezione della Gestione Anagrafiche è simile a quella di figura 22;

Schermata di Gestione delle Anagrafiche

La prima riga presenta la funzionalità *filtro di ricerca*; basterà specificare i valori conosciuti dell'elemento che si desidera ottenere, anche parziali, e la lista successiva presenterà solo gli elementi che corrispondono a tali informazioni. Ogni riga della lista rappresenta un elemento dell'anagrafica; è composta (sempre se i permessi associati all'utente in uso lo consentono) da tre tasti rappresentanti in ordine le funzionalità di *visualizzazione*, *modifica e cancellazione* e successivamente dai campi relativi alle informazioni principali riguardanti l'elemento. Infine si ha un tasto di *aggiunta* degli elementi per completare le funzionalità.

Alcune delle funzionalità menzionate faranno comparire un modulo popup, eventualmente precompilato con tutti i dati relativi all'elemento scelto, per il quale, esclusa la funzione di visualizzazione, sarà concessa la modifica e il successivo invio. I campi a compilazione obbligatoria sono denotati con un asterisco; qualora uno di questi non fosse compilato, oppure in generale un campo non venga compilato con il tipo di dato corrispondente, il tentativo di invio fallirà e sarà richiesta la correzione, evidenziando i campi dove questo risulti necessario.

Modulo popup delle Anagrafiche

L'interfaccia della schermata di Gestione delle Offerte è strutturata in modo analogo a quella delle anagrafiche; presenta quindi la funzionalità filtro, nonché quelle di visualizzazione, modifica, eliminazione e creazione delle offerte.

Schermata di Gestione delle Offerte

Il modulo che compare per la modifica e la creazione delle offerte, per la complessità delle informazioni che gestisce, è stato suddiviso in tre sezioni:

Modulo popup delle Offerte

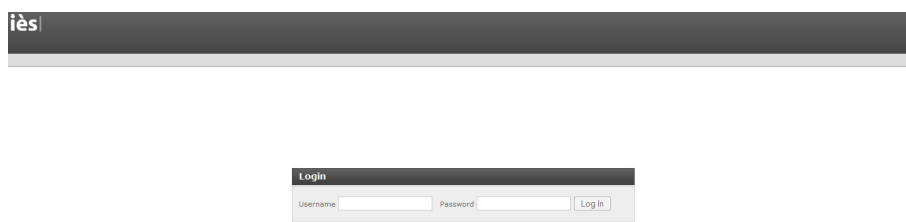


Figura 21: Schermata di login

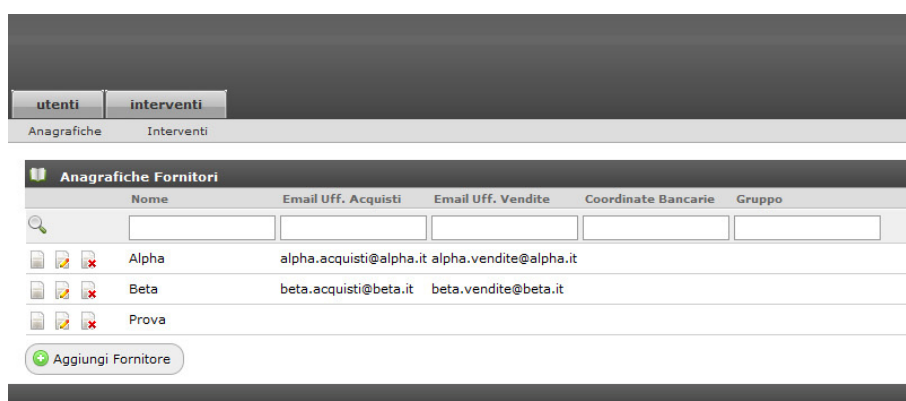


Figura 22: Schermata delle anagrafiche

The screenshot shows a modal form titled "Fornitore". It has a title bar with a close button (X). The form contains several input fields: "Nome*", "Email Uff. Acquisti:", "Email Uff. Vendite:", "Coordinate Bancarie:", and "Gruppo:". Below these fields is a "Note:" field with an "Aggiungi Nota" button. At the bottom left, there is a "Salva" button.

Figura 23: Form relativo all'anagrafica dei Fornitori

ID Offerita	Cliente	Stato	Preventivo	Data inizio offerta	Data scadenza offerta	Data conferma ordine	Data scadenza ordine
123	CLIENTE GENERICO PER	Preventivo	1234€	06/07/2011	06/07/2011	06/07/2011	06/07/2011
5	CLIENTE GENERICO PER	Scaduto	343.67€	28/04/2011	28/04/2011	28/04/2011	28/04/2011
253	CONDOMINIO GENERICO PER	Scaduto	23€	29/04/2011	30/04/2011	29/04/2011	29/04/2011
4	CORRISPETTIVO	Scaduto	34.99€	05/04/2011	05/04/2011	13/04/2011	29/04/2011
3	CORRISPETTIVO	Scaduto	34.6€	28/04/2011	28/04/2011	28/04/2011	28/04/2011

Figura 24: Schermata di gestione delle Offerte

una sezione principale, una dei luoghi e una delle procedure. La sezione principale è quella che viene proposta inizialmente; qui è possibile operare sui dati strettamente riguardanti l'offerta, come il cambio di stato (quando concesso), la formulazione di preventivi, la modifica delle date di scadenza. E' concessa in fase di formulazione del preventivo inoltre la scelta di un contatto o cliente, selezionabile da un menù a tendina; qualora non fosse ancora disponibile, basterà selezionare la voce 'nuovo contatto' nel menù a tendina per fare apparire un modulo extra, dotato di tutti i campi necessari per l'inserzione del nuovo destinatario. La seconda sezione, quella dei luoghi, fornisce un menù di selezione che propone i luoghi collegati al destinatario selezionato nella fase precedente; è possibile effettuare selezioni multipli qualora il contratto dovesse riguardare operazioni da effettuare su più di un luogo. Inoltre se il luogo non dovesse essere presente nella lista proposta anche in questo contesto viene concessa la possibilità di aggiungere moduli extra per l'inserzione dei dati relativi ai nuovi luoghi che vengono presi in considerazione. L'ultima sezione, delle procedure, consente infine di assegnare ai luoghi specificati in precedenza le tipologie di intervento che vi si intende eseguire, selezionabili anch'esse da menù a tendina; per ogni tipologia di intervento aggiunta compare un nuovo modulo, il quale richiede dati (per quelle che poi a livello logico diventeranno le Sessioni) come il numero di interventi da eseguire, la loro durata, il numero di giorni da attendere tra un intervento e l'altro, un eventuale periodo in cui devono essere svolti e, sempre eventualmente, la definizione dei vincoli di operatore, di mezzo e di attrezzatura.

La schermata di Gestione del Calendario si propone inizialmente su base mensile. E' possibile navigare tra i vari mesi con due tasti di scorrimento in cima al calendario. Il mese è ordinato per righe settimanali, ed ogni riga è espandibile per poter visualizzare i piani orari dei giorni che la compongono; ogni ora lavorativa è seguita dalla lista degli interventi che vi sono schedati, e presenta un indicatore dell'occupazione a fianco, che indica il numero di operatori utilizzati rispetto a quelli disponibili.

Schermata mensile

Viene offerta in alto alla schermata la possibilità di lanciare la funzionalità di schedulazione automatica cliccando sul tasto corrispondente, che dopo un breve periodo di elaborazione rimanda a questa schermata, dove è possibile controllare le proposte di programma.

Ogni intervento è notato sul calendario con il nome del cliente su cui va ad agire. Scorrendo il mouse su un intervento singolo sul calendario compare un riquadro temporaneo che ne ricapitola le informazioni principali; cliccando

Offerta ✕

Principale Luoghi Procedure


ID Offerta:


Tacito Rinnovo*: Sì No


Intestatario*: -- Scegli un intestatario -- ▼


Stato: Preventivo ▼

Preventivo*: €

Data Inizio Offerta*: 

Data Scadenza Offerta*: 

Data Conferma Ordine: 

Data Scadenza Ordine: 

Note:

Figura 25: Form delle Offerte, modulo principale

Offerta [X]

Principale **Luoghi** Procedure

Aggiungi Esistente: -- Scegli un luogo -- [v] Agg. Luogo

Luoghi: - via Venezia 1, Padova (PD) [X]

Aggiungi Nuovo:

Nuovo Luogo #1 [↑] [X]

Nome: []

Telefono: []

Riferimento: []

Regione: Abruzzo [v]

Provincia: Chieti (CH) [v]

Comune*: []

Zona: [v]

Indirizzo*: []

N. Civico*: []

Indicazioni: []

Limiti Accessibilità: Lun Mar Mer Gio Ven Sab Dom

Note: [Aggiungi Nota]

Nuovo Luogo

Figura 26: Form delle Offerte, modulo dei luoghi

Offerta ✕

Principale
Luoghi
Procedure

Luogo: - via Venezia 1, Padova (PD)
↑ ✕

#1 - Zanzara: Adulticida
↑ ✕

N. Interventi*:

Intervallo Interventi*:

Durata Intervento*:

Data Inizio:

Data Fine:

Vincoli Operatori: -- Scegli un operatore -- Agg.

Vincoli Mezzi: Prova Si No ✕
 -- Scegli un mezzo -- Agg.

Vincoli Attrezzature: -- Scegli un attrezzo -- Agg.

Note: Aggiungi Nota

Figura 27: Form delle Offerte, modulo delle procedure

Calendario

Lancia Scheduling Automatica

◀ **Luglio 2011** ▶

Luglio 2011						
Domenica	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato
					1	2
					Venerdì	Sabato
3	4	5	6	7	8	9
08:00 - []	08:00 - []	08:00 - []	08:00 - []	08:00 - []	08:00 - []	08:00 - []
08:30 - []	08:30 - []	08:30 - []	08:30 - []	08:30 - []	08:30 - []	08:30 - []
09:00 - []	09:00 - []	09:00 - []	09:00 - []	09:00 - [1]	09:00 - []	09:00 - []
09:30 - []	09:30 - []	09:30 - []	09:30 - []	CLIENTE GENERICO	09:30 - []	09:30 - []
10:00 - []	10:00 - []	10:00 - []	10:00 - []	09:30 - [1]	10:00 - []	10:00 - []
10:30 - []	10:30 - []	10:30 - []	10:30 - []	10:00 - []	10:30 - []	10:30 - []
11:00 - []	11:00 - []	11:00 - []	11:00 - []	10:30 - []	11:00 - []	11:00 - []
11:30 - []	11:30 - []	11:30 - []	11:30 - []	11:00 - []	11:30 - []	11:30 - []
12:00 - []	12:00 - []	12:00 - []	12:00 - []	11:30 - []	12:00 - []	12:00 - []
12:30 - []	12:30 - []	12:30 - []	12:30 - []	12:00 - []	12:30 - []	12:30 - []
13:00 - []	13:00 - []	13:00 - []	13:00 - []	12:30 - []	13:00 - []	13:00 - []
13:30 - []	13:30 - []	13:30 - []	13:30 - []	13:00 - [2]	13:30 - []	13:30 - []
14:00 - []	14:00 - []	14:00 - []	14:00 - []	CONDOMINIO GEN.	14:00 - []	14:00 - []
14:30 - []	14:30 - []	14:30 - []	14:30 - []	13:30 - [2]	14:30 - []	14:30 - []
15:00 - []	15:00 - []	15:00 - []	15:00 - []	14:00 - [2]	15:00 - []	15:00 - []
15:30 - []	15:30 - []	15:30 - []	15:30 - []	14:30 - [2]	15:30 - []	15:30 - []
16:00 - []	16:00 - []	16:00 - []	16:00 - []	15:00 - []	16:00 - []	16:00 - []
16:30 - []	16:30 - []	16:30 - []	16:30 - []	15:30 - []	16:30 - []	16:30 - []
17:00 - []	17:00 - []	17:00 - []	17:00 - []	16:00 - []	17:00 - []	17:00 - []
17:30 - []	17:30 - []	17:30 - []	17:30 - []	16:30 - []	17:30 - []	17:30 - []
18:00 - []	18:00 - []	18:00 - []	18:00 - []	17:00 - []	18:00 - []	18:00 - []
18:30 - []	18:30 - []	18:30 - []	18:30 - []	17:30 - []	18:30 - []	18:30 - []
				18:00 - []		
				18:30 - []		
10	11	12	13	14	15	16
Domenica	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato
17	18	19	20	21	22	23
Domenica	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato
24	25	26	27	28	29	30
Domenica	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato
31						

Figura 28: Schermata mensile di Gestione del Calendario

Modifica Intervento

ID Offerta: OC-123
 Cliente: CLIENTE GENERICO PER ZANZARE
 Data Scadenza: 06/07/2011
 Sessione: 1
 Procedura: Adulticida (Zanzara)
 Luogo: via Venezia 1, Padova (PD) Zona:
 Periodo: 06/07/2011 - 28/07/2011
 Intervento: 1/2
 Data: 07/07/2011
 Ora: 09:00
 Durata: 1
 Operatori: Giovanni Bianchi
 Mezzi: Prova
 Catture:
 Report: Sfoglia...
 Successo:
 Note: Aggiungi Nota
 Salva

Figura 29: Form di modifica degli Interventi

invece sull'intervento singolo viene aperto il modulo di modifica, che permette di visualizzare i dettagli relativi al contratto di cui fa parte, il cliente per cui viene eseguito, la tipologia di intervento che va a svolgere e il relativo luogo, e di cambiarne i dati relativi a data e ora di svolgimento, durata, e risorse impiegate, nonché di comunicare il successo e caricare un file di report a intervento compiuto.

Modulo di modifica degli interventi

Cliccando invece dalla schermata mensile sulle date del calendario è possibile passare alla loro gestione giornaliera; in questo contesto abbiamo una visualizzazione tabulare del programma, con le ore lavorative disposte sulle righe, gli operatori disponibili sulle colonne e gli interventi collocati nelle intersezioni tra di essi; se ad un intervento dovesse essere associato a più di un operatore, il suo riquadro sarà collocato in tutte le colonne richieste; i riquadri multipli saranno comunque riconducibili allo stesso intervento grazie a un codice di colori, che associando lo stesso colore a più riquadri dello stesso intervento li rende subito riconoscibili. E' inoltre associato un codice ai riquadri degli interventi stessi per permettere di riconoscere immediatamente anche il loro stato di schedulazione: le schedulazioni o hanno quindi

Schermata giornaliera

Giornaliero
 Giovedì 7 Luglio 2011

	Giovanni Bianchi	Filippo Rossi	Michele Verdi
08:00			
08:30			
09:00	<input type="checkbox"/> CLIENTE GENERICO		
09:30			
10:00			
10:30			
11:00			
11:30			
12:00			
12:30			
13:00		<input checked="" type="checkbox"/> CONDOMINIO GEN.	<input checked="" type="checkbox"/> CONDOMINIO GEN.
13:30			
14:00			
14:30			
15:00			
15:30			
16:00			
16:30			
17:00			
17:30			
18:00			
18:30			

STAMPA STAMPA STAMPA

Schedulabili Riempi

Nessun intervento.

Figura 30: Schermata giornaliera di Gestione del Calendario

un riquadro tratteggiato, le prenotazioni uno continuo ma sottile, mentre le conferme avranno un riquadro marcato. Per facilitare l'individuazione degli orari disponibili durante la schedulazione manuale inoltre le celle orarie occupate vengono marcate in grigio.

Su questa schermata è possibile praticare le operazioni di schedulazione manuale come menzionato nel capitolo precedente; è quindi disponibile una barra degli interventi schedulabili in grado di proporre gli interventi disponibili per la programmazione in giornata, le operazioni di trascinamento per la modifica veloce degli orari e degli operatori e una schermata di modifica dettagliata come quella presente per la schermata mensile.

Stampa del programma

E' infine possibile effettuare un riepilogo del programma di ogni operatore grazie al pulsante 'Stampa' collocato in fondo a ciascuna colonna. Qui viene infatti proposto un popup dotato di tale riepilogo composto dalle informazioni necessarie a ciascun operatore per agire, con la possibilità di effettuare la stampa.

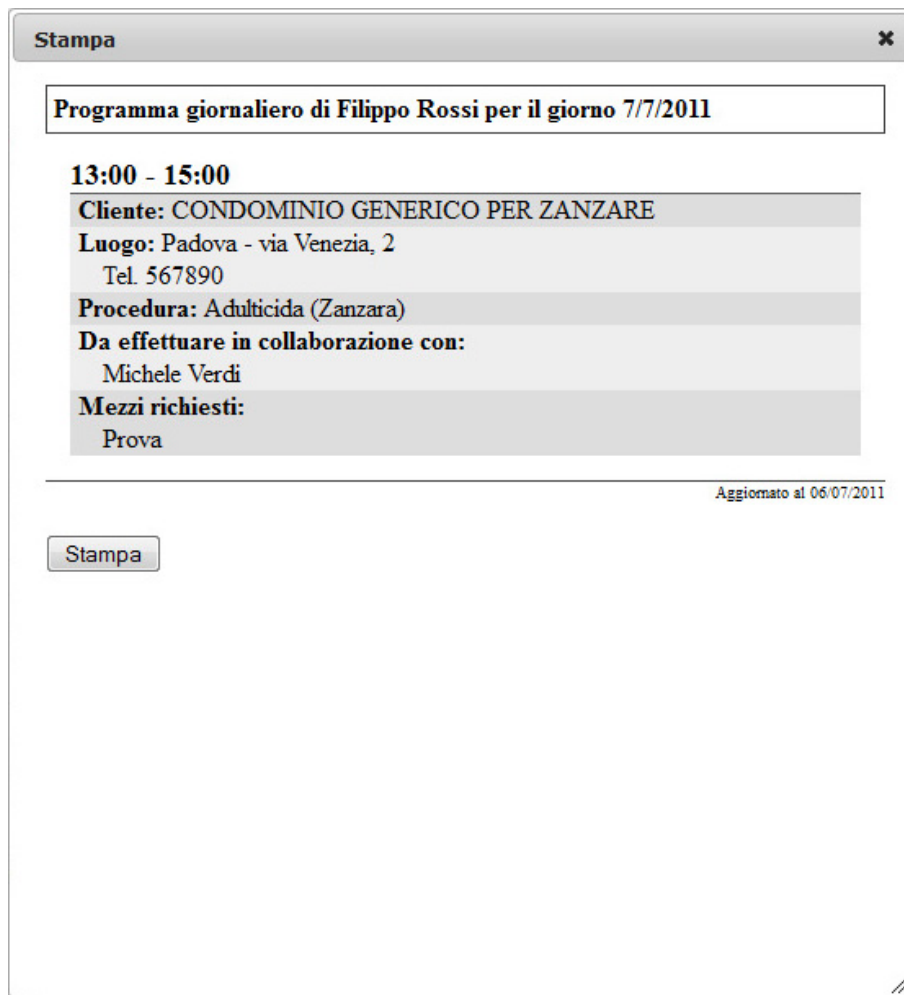


Figura 31: Stampa del programma giornaliero

6

CONCLUSIONI

INDICE

6.1	Sviluppi futuri	65
6.2	Considerazioni sull'esperienza	65
6.3	Ringraziamenti	66

6.1 SVILUPPI FUTURI

Il software applicativo descritto in questo testo è già funzionale e in grado di applicarsi a casi d'uso reali, tuttavia è in grado di sviluppare ampi margini di miglioramento. In primo luogo è prevista l'aggiunta di ulteriori servizi; alcuni esempi notabili sono una gestione dei pagamenti ed un servizio di documentazione in grado di consentire l'invio e l'organizzazione di file di rapportistica e foto collegati alle voci della base di dati. L'algoritmo di risoluzione del problema del cammino a costo minimo sarà migliorato attraverso la funzionalità di geolocalizzazione: è prevista l'aggiunta di funzioni di collegamento con il servizio Google Maps, in modo da permettere la traduzione automatizzata degli indirizzi inseriti nelle coordinate geografiche corrispondenti, utilizzate successivamente per l'algoritmo risolutivo. Tale servizio può permettere inoltre di aggiungere informazione alle stampe dei piani giornalieri per gli operatori; questi infatti potranno essere dotati di carte stradali in grado di indicare il percorso dettagliato da seguire per arrivare dal luogo di un intervento al successivo. Infine, una volta ultimato, il sistema sarà adattabile all'applicazione in contesti diversi da quello della disinfestazione: la natura gestionale e le problematiche comuni (come appunto il problema TSP) trattati rendono questo possibile.

Aggiunta di funzionalità

Geocoding

Generalizzazione della problematica

6.2 CONSIDERAZIONI SULL'ESPERIENZA

Durante lo svolgimento del tirocinio mi è stato dato modo di applicare molti dei concetti appresi nel corso di studi, come ad esempio la progettazione di basi di dati o gli algoritmi analizzati dalla ricerca operativa, portando le mie conoscenze nella metodologia di lavoro adottata alla Sphera Engineering Communication. Ho avuto inoltre l'occasione di riprendere e ampliare le mie conoscenze pregresse nel campo dello sviluppo web con linguaggi come HTML, CSS e Javascript, con cui avevo già lavorato nell'ambito professionale, nonché di imparare l'uso di PHP e adottare librerie di programmazione di cui avevo solamente sentito parlare in precedenza come jQuery e Doctrine ORM. La maggiore difficoltà affrontata è stata principalmente quella causata dal dover sviluppare in tempo breve il modulo applicativo, dovendo alternarne lo sviluppo con lo studio delle conoscenze teoriche necessarie ad esso. A

Concetti appresi

Difficoltà riscontrate

causa di motivi logistici non è stato infatti possibile portare avanti una formazione completa prima dell'inizio dello sviluppo, e questo ha portato ad un rallentamento ed a dover tralasciare alcune delle funzionalità sopra menzionate. In particolare tale difficoltà si è presentata nella fase iniziale del tirocinio, quando le conoscenze di PHP e dell'interfacciamento con il sistema IES erano ancora poco sviluppate. Tuttavia il tirocinio si è svolto con reciproca soddisfazione mia e dell'azienda, e si concluso con la proposta di proseguire il rapporto continuando lo sviluppo del modulo e formandomi maggiormente sul sistema IES, per divenire maggiormente coinvolto nello sviluppo delle relative applicazioni.

6.3 RINGRAZIAMENTI

Intendo rivolgere un sentito ringraziamento verso le seguenti persone: Il mio tutore e relatore, il professor Marcello Dalpasso per avermi seguito con pazienza e per il prezioso aiuto offerto durante la stesura di questo documento; Il mio tutore aziendale Andrea Secchi per avermi accolto e seguito nel corso del tirocinio e i colleghi di Sphera; La mia famiglia, per avermi offerto l'occasione di portare a termine questo corso di laurea e avermi dato sostegno in questi anni; I miei amici Luca, Alessandro, Alberto e Francesco, nonché i compagni di studio in generale della sede di Treviso.

BIBLIOGRAFIA

- [1] Fischetti, Matteo (1995), *Lezioni di Ricerca Operativa*, Edizioni Libreria Progetto, Padova
- [2] "PHP" Wikipedia, L'enciclopedia libera. 20 mag 2011, 10:00 <<http://it.wikipedia.org/wiki/PHP>>
- [3] "HTML" Wikipedia, L'enciclopedia libera. 20 mag 2011, 10:00 <<http://it.wikipedia.org/wiki/HTML>>
- [4] "CSS" Wikipedia, L'enciclopedia libera. 20 mag 2011, 10:00 <<http://it.wikipedia.org/wiki/CSS>>
- [5] "Javascript" Wikipedia, L'enciclopedia libera. 20 mag 2011, 10:00 <<http://it.wikipedia.org/wiki/JavaScript>>