

Claudio Gecchele 578404-IF

Università di Padova  
Facoltà di Ingegneria

## **Progettazione e sviluppo di un modulo per la georeferenziazione di dati aziendali**



Studente: Claudio Gecchele  
Relatore: Prof. Massimo Rumor

## Indice

1.	Abitat SIT srl .....	5
1.1	L'azienda.....	5
1.2	Le competenze e le abilità operative .....	5
1.2.1	Utility ed applicazioni.....	5
1.2.2	Cartografia catastale .....	5
1.2.3	Formazione .....	5
1.2.4	Consulenza .....	5
1.2.5	Supporto tecnico.....	6
1.2.6	Ortofotocarte.....	6
1.2.7	Rilievi e censimenti.....	6
1.2.8	Informatizzazione delle reti tecnologiche.....	6
2.	Sistema Informatico aziendale.....	6
2.1	SugarCRM.....	6
2.1.1	Introduzione.....	6
2.1.2	Anagrafiche Clienti.....	7
3.	Obiettivi del tirocinio .....	7
4.	Tecnologia usata .....	8
4.1	PHP.....	8
4.2	Javascript.....	8
4.3	PostgreSQL.....	8
4.4	OSGeo MapGuide .....	9
4.4.1	MapGuide Server .....	10
4.4.2	MapGuide Web Server Extensions .....	10
4.4.3	Autodesk MapGuide <sup>TM</sup> Studio e MapGuide Maestro .....	10
4.5	Eclipse.....	10
4.6	Google Maps .....	10
4.6.1	Uso delle Geocoding API.....	10

4.6.2	Costi e limiti delle Geocoding API.....	11
4.7	Bing Maps.....	11
4.7.1	Uso delle Location API.....	11
4.7.2	Costi e limiti delle Location API.....	11
4.8	Tele Atlas.....	11
5.	Lavoro svolto.....	12
5.1	Introduzione .....	12
5.1.1	Framework AbFmw.....	12
5.1.2	Platform Land® versione base.....	12
5.2	Piano di progetto.....	12
5.3	Formazione.....	13
5.3.1	OSGeo MapGuide .....	13
5.3.2	Platform Land® .....	13
5.4	Analisi dei Requisiti .....	13
5.4.1	Casi d'uso .....	13
5.4.2	Tabella dei requisiti.....	14
5.4.3	Tabella di tracciamento .....	15
5.5	Progettazione.....	16
5.5.1	Definizione del prodotto .....	16
5.5.2	Descrizione dei componenti .....	16
5.5.3	Diagrammi di attività.....	18
5.5.4	Design Pattern .....	20
5.5.5	Tracciamento requisiti – componenti.....	20
5.5.6	Tracciamento requisiti – Metodi.....	21
5.6	Realizzazione e Collaudo.....	22
5.6.1	Inserimento dei clienti in mappa.....	22
5.6.2	Ricerca per comune, per comune e indirizzo, per comune, indirizzo e civico e ricerca per cliente .....	35

6.	Risultati raggiunti.....	76
6.1	Funzionalità di inserimento dati clienti.....	76
6.2	Ricerche .....	78
6.3	Tematiche statiche per settore, tipo, utente assegnato, categoria .....	78
7.	Sviluppi previsti .....	79
7.1	Miglioramenti .....	79
7.2	Nuove funzionalità .....	79
7.2.1	Filtri dinamici.....	79
7.2.2	Inserimento di ulteriori dati.....	79
	Figura 1: Parte utilizzata della base di dati del crm.....	7
	Figura 2: Schema architetturale di OSGeo MapGuide .....	9
	Figura 3: Schema uml delle classi progettate .....	18
	Figura 4: Diagramma di Attività per l’algoritmo di geocodifica.....	19
	Figura 5: Interfaccia per l’inserimento dei dati dei clienti.....	76
	Figura 6: Caricamento dei dati in esecuzione.....	77
	Figura 7: Caricamento dei dati eseguito .....	78

## **1. Abitat SIT srl**

### **1.1 L'azienda**

Abitat SIT è nata nel 1983 come società di progettazione e consulenza in ambito civile e industriale e dal 1992 è impegnata con la divisione Sistemi Informativi Territoriali nella fornitura di servizi e produzione di software dedicato principalmente alla risoluzione di problematiche di gestione del territorio. Nel corso degli anni ha esteso la gamma dei servizi a disposizione per completare il supporto fornito ai propri utenti. Inoltre collabora in sinergia con studi professionali che apportano nuove conoscenze e specializzazioni in campi che interagiscono con i Sistemi Informativi Territoriali.

### **1.2 Le competenze e le abilità operative**

#### **1.2.1 Utility ed applicazioni**

L'azienda è stata in grado di sviluppare una sensibilità notevole verso i problemi affrontati cercando di semplificare le modalità di gestione dei progetti. Inoltre grazie ad un grande lavoro di apprendimento e conoscenza delle problematiche dell'utente finale i prodotti realizzati rispondono alle esigenze operative richieste mantenendo una rilevante semplicità d'uso.

#### **1.2.2 Cartografia catastale**

La possibilità di entrare in possesso, da parte delle agenzie territoriali, dei fogli catastali in formato vettoriale ha consentito all'azienda di attivare nuove funzionalità informatiche per utilizzare i dati catastali. Tali funzionalità permettono all'utente di utilizzare una base cartografica tecnologicamente pronta per essere recepita dai software CAD e GIS per la visualizzazione in livelli multipli e sovrapponibili.

#### **1.2.3 Formazione**

Abitat SIT, allo scopo di far ottimizzare gli investimenti intrapresi ai propri clienti organizza corsi di formazione all'uso dei software commercializzati, sia su argomenti di base, sia su argomenti specifici.

#### **1.2.4 Consulenza**

L'azienda fornisce servizi di consulenza ad aziende pubbliche o private che hanno bisogno di redigere particolare documentazione o di organizzare progetti che richiedono personale specializzato.

### **1.2.5 Supporto tecnico**

Abitat SIT presta servizi di assistenza telefonica, di intervento o remota su tutti i suoi prodotti a seconda dei contratti stipulati per dare un importante sostegno al cliente che ha difficoltà a correggere malfunzionamenti o eseguire determinate operazioni in completa autonomia.

### **1.2.6 Ortofotocarte**

L'azienda è in grado di fornire ortofotocarte realizzate attraverso l'uso di tecnologie sofisticate e processi a scansione digitale con precisione e qualità migliori di quelle realizzate con sistemi puramente geometrici. Inoltre può fornire ortofotocarte satellitari elaborate ed inserite in mappe GIS in sovrapposizione con altri livelli cartografici.

### **1.2.7 Rilievi e censimenti**

Abitat SIT può svolgere servizi di rilievo e censimento di varie entità come la numerazione civica, l'illuminazione pubblica, le schedature di edifici, la cartellonistica stradale e pubblicitaria. L'informatizzazione dei dati raccolti viene fatta dall'azienda stessa oppure può essere eseguita internamente dall'utente dopo essersi dotato della strumentazione informatica necessaria.

### **1.2.8 Informatizzazione delle reti tecnologiche**

Abitat SIT ha esperienza nell'informatizzazione di cartografie che rappresentano reti tecnologiche, fornendo funzionalità di georeferenziazione del dato, di ricerca in base a criteri specifici, di stampa in vari formati e di inserimento di nuove informazioni.

## **2. Sistema Informatico aziendale**

Il sistema informatico aziendale è composto da cinque divisioni: amministrativa, commerciale, sviluppo software, locale server. Il modulo software sviluppato potrà essere utilizzato dalla divisione commerciale in quanto amplia le funzionalità messe a disposizione dal software CRM in uso (SugarCRM), consentendo di prendere decisioni in base al territorio.

### **2.1 SugarCRM**

#### **2.1.1 Introduzione**

SugarCRM è un software open source molto diffuso che si occupa della gestione delle informazioni generate dall'azienda nei rapporti con i propri clienti o potenziali clienti. Si tratta di un'applicazione web-based molto flessibile e adattabile alle specifiche esigenze dell'azienda. In particolare permette di gestire in modo completo:

- le anagrafiche clienti, i contatti e i potenziali clienti
- la forza vendita aziendale
- le campagne di marketing
- i ticket di supporto e il knowledge base aziendale
- la registrazione sul cliente delle attività ordinarie, come chiamate, lavori, appuntamenti o documentazione
- progetti aziendali e gruppi di lavoro

### 2.1.2 Anagrafiche Clienti

Le anagrafiche clienti contengono le informazioni basilari per poter gestire i rapporti con i clienti. In particolare per sviluppare questa prima versione dell'applicazione è stata utilizzata la parte di base di dati indicata nello schema concettuale sottostante.

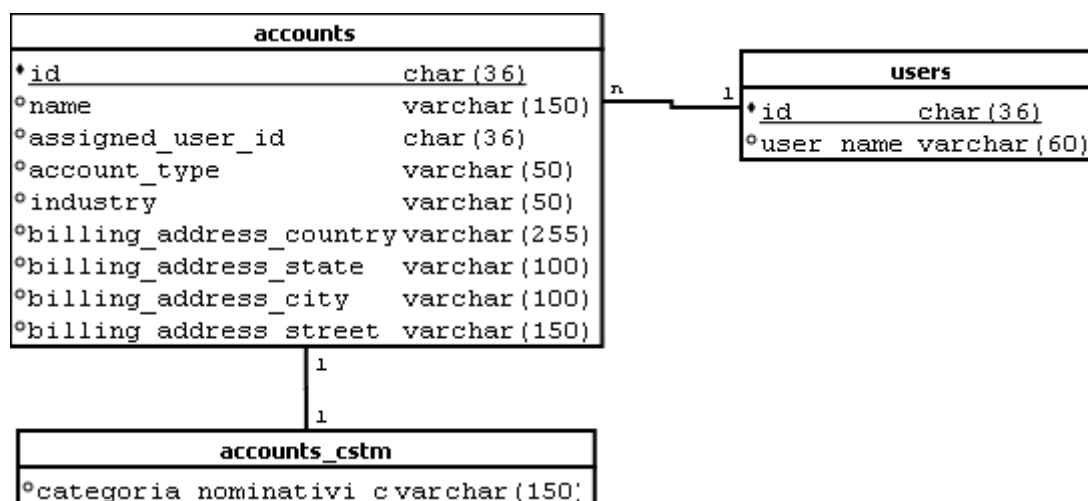


FIGURA 1: PARTE UTILIZZATA DELLA BASE DI DATI DEL CRM.

## 3. Obiettivi del tirocinio

Il tirocinio ha come obiettivo lo sviluppo di un'applicazione che consenta di geolocalizzare le informazioni riguardanti i clienti presenti nel CRM, visualizzarli e analizzarli attraverso funzioni di ricerca e di stampa. In particolare in questa prima versione si vogliono sviluppare le seguenti funzionalità:

- Inserimento dei clienti in mappa
- Visualizzazione delle geometrie relative ai clienti
- Visualizzazione dei dati dei clienti

- Ricerca per comune
- Ricerca per comune, via
- Ricerca per comune, via, civico
- Ricerca per cliente
- Filtri statici sui dati dei clienti
- Stampa

Inoltre l'applicazione deve essere il più possibile indipendente dall'applicazione CRM dalla quale vengono ottenuti i dati aziendali e dalla tecnologia utilizzata per la geocodifica degli stessi.

## **4. Tecnologia usata**

### **4.1 PHP**

PHP (PHP Hypertext Preprocessor) è un linguaggio di scripting lato server interpretato, utilizzato inizialmente per lo sviluppo di pagine web dinamiche. Attualmente viene utilizzato per sviluppare intere applicazioni lato server, scrivere script a riga di comando e applicazioni stand-alone con interfaccia grafica. Inoltre è un linguaggio orientato agli oggetti e dotato di una vastissima libreria che contiene funzionalità complesse come l'accesso ai database, la manipolazione d'immagini o la compressione/decompressione dati. Nel progetto è stato utilizzato per realizzare le componenti del software che devono essere eseguite sul server.

### **4.2 Javascript**

Javascript è un linguaggio di scripting lato client interpretato dotato di tutte le caratteristiche necessarie per implementare algoritmi strutturati. All'interno del progetto è stato utilizzato per rendere interattiva l'applicazione senza essere costretti ad aggiornare tutta la pagina, ma solamente gli elementi modificati, anche attraverso l'uso della tecnica di sviluppo AJAX (Asynchronous Javascript and XML).

### **4.3 PostgreSQL**

PostgreSQL è un R-DBMS cioè un sistema di gestione di basi di dati relazionale. Nel progetto è stato utilizzato per gestire la base di dati della versione base di Platform Land® (modificata in alcune sue parti) e quella relativa ai dati Tele Atlas.



#### 4.4 OSGeo MapGuide

OSGeo MapGuide è una piattaforma web-based open source che permette agli utenti di sviluppare applicazioni di web mapping e web services geospaziali; utilizza un'architettura a tre livelli: livello server, livello web e livello client, per ognuno dei quali si possono trovare diverse componenti ciascuna con il proprio compito.

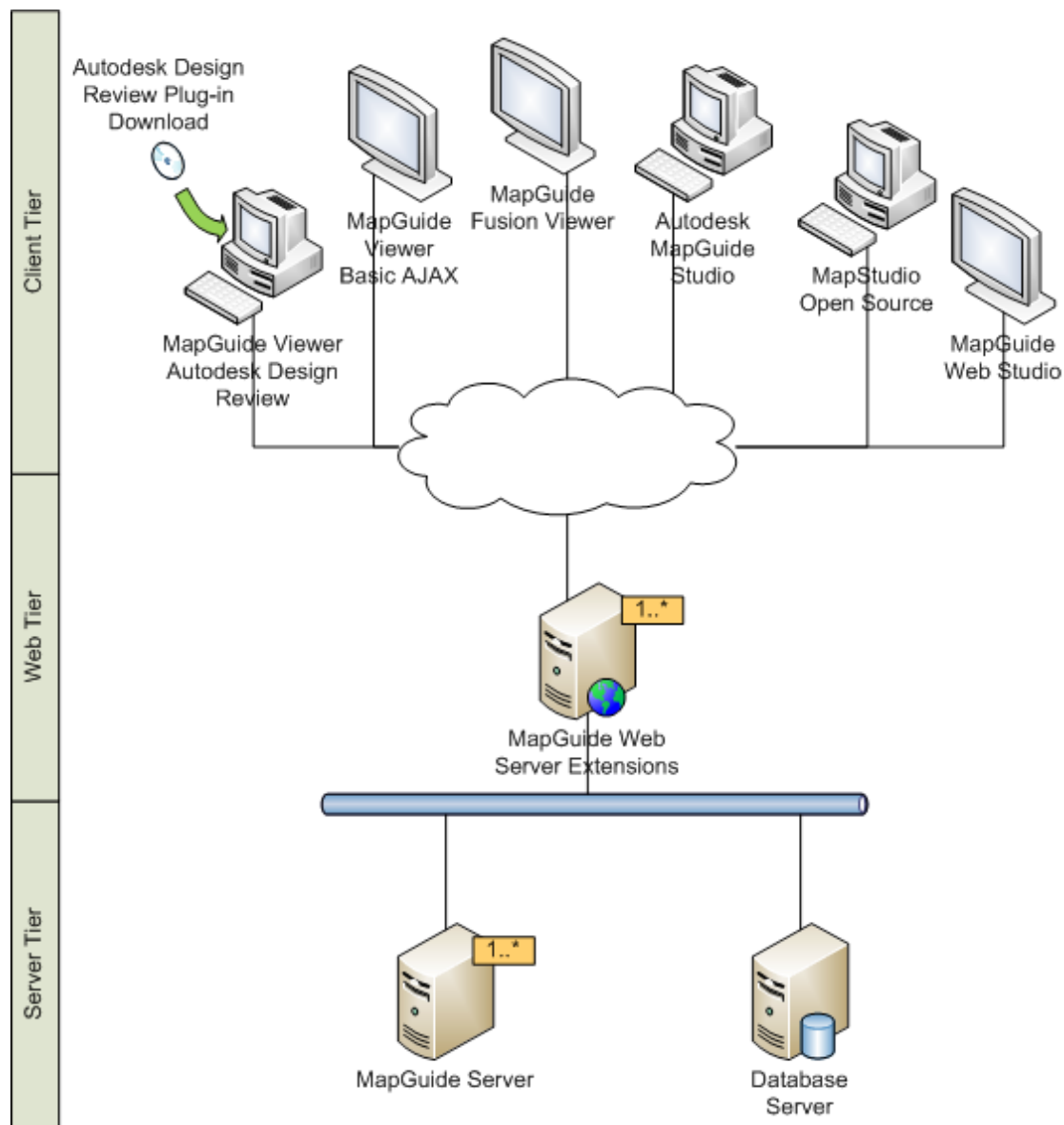


FIGURA 2: SCHEMA ARCHITETTURALE DI OSGEO MAPGUIDE

#### **4.4.1 MapGuide Server**

MapGuide Server risponde alle richieste delle applicazioni client attraverso il protocollo TCP/IP e fornisce sette servizi: Site Service, Resource Service, Drawing Service, Feature Service, Mapping Service, Rendering Service, Tile Service e KML Service.

#### **4.4.2 MapGuide Web Server Extensions**

Le Mapguide Web Server Extensions sono componenti interne alle quali si accede attraverso la MapGuide Web API perché non presentano un'interfaccia grafica. Queste estensioni hanno il compito di esporre i servizi offerti dal MapGuide server alle applicazioni client utilizzando il protocollo HTTP. Tra queste si trova l'applicazione MapGuide Site Administrator che permette di gestire e configurare i server e definire degli alias mnemonici per i percorsi dei file che vengono utilizzati nelle applicazioni create.

#### **4.4.3 Autodesk MapGuide™ Studio e MapGuide Maestro**

Autodesk MapGuide Studio e MapGuide Maestro (il primo proprietario, il secondo open source) permettono di caricare e configurare le sorgenti di dati spaziali, di definire i layer, le mappe e gli elementi dell'interfaccia utente che devono essere visualizzati dall'applicazione.

### **4.5 Eclipse**

Eclipse è un ambiente di sviluppo integrato multiplatforma perché scritto in linguaggio Java e multi-linguaggio perché permette di sviluppare software in linguaggio Java, C++ e, attraverso plug-in specifici, XML, Javascript e PHP. Nel progetto è stato utilizzato per sviluppare l'applicazione con i linguaggi PHP e Javascript e per effettuare test sulle parti in linguaggio PHP.

### **4.6 Google Maps**

Google Maps mette a disposizione due tipi di servizi gratuiti che, tra le altre funzionalità, permettono di effettuare la geocodifica: Javascript API V3 Client Geocoder che è più adatto ad eseguire la geocodifica dinamica, cioè in presenza di un'interfaccia grafica, e le Google Geocoding API che consentono di eseguire la geocodifica statica, cioè quando i parametri utilizzati (stato, paese, ecc) provengono da una base di dati e non, ad esempio, da un form inseriti dall'utente.

#### **4.6.1 Uso delle Geocoding API**

Per utilizzare le Geocoding API è sufficiente richiamare un url specificando il formato in cui deve essere restituito il risultato (json o xml), i parametri che servono per eseguire

la geocodifica e il parametro sensor (true o false), che indica se chi fa la richiesta è dotato di un sensore di posizione. Se viene scelto come formato di output l'xml, basta utilizzare la funzione `php simplexml_load_file()` che permette di caricare un file in formato xml dato l'url del file.

#### **4.6.2 Costi e limiti delle Geocoding API**

A causa dei limiti non molto chiari inerenti all'uso delle Geocoding API e dei costi derivanti dall'acquisto delle Google Maps API Premier, qualora fosse necessario, si è deciso di non utilizzarle per lo sviluppo dell'applicazione, ma di mantenere la compatibilità per un eventuale uso futuro.

### **4.7 Bing Maps**

Bing Maps mette a disposizione il servizio Location API, facente parte delle Bing Maps REST Services API, che, come per le Geocoding API di Google Maps, consente di eseguire la geocodifica di un indirizzo. Ci sono diverse versioni a seconda degli scopi per i quali viene utilizzato: si passa dalla versione base per gli sviluppatori alla versione a pagamento che non ha limiti.

#### **4.7.1 Uso delle Location API**

Per usare le Location API basta richiamare un url specificando i parametri relativi all'indirizzo, il formato di output del risultato (json o xml) e una chiave denominata BingMapsKey e ottenibile in fase di registrazione al servizio in base alla versione scelta.

#### **4.7.2 Costi e limiti delle Location API**

Per gli stessi motivi visti per i servizi offerti da Google Maps si è deciso di non utilizzare tale servizio, per il momento, e di mantenere la compatibilità per un probabile uso futuro non appena saranno chiariti i costi e i limiti.

### **4.8 Tele Atlas**

Tele Atlas ha prodotto la piattaforma Multinet, un database che contiene tutte le informazioni alfanumeriche e cartografiche della geografia mondiale permettendo di realizzare una mappa digitale del mondo. Utilizzando tali dati è possibile sviluppare un algoritmo di geocodifica personale senza l'uso di servizi esterni, anche se potrebbe richiedere un notevole impiego di tempo. Per questo e per il fatto che tale tecnologia era già stata utilizzata in azienda e se ne conoscevano i costi si è deciso di basare su questa piattaforma, questa prima versione del prodotto.

## **5. Lavoro svolto**

### **5.1 Introduzione**

Dato il tempo a disposizione, la difficoltà nel progettare l'algoritmo di geocodifica e le numerose tecnologie da integrare ho deciso di sviluppare per questa prima versione un prototipo a partire da un software (Platform Land®) già realizzato, che è in grado di svolgere buona parte delle funzionalità richieste, e da un framework che consente di semplificarne lo sviluppo. Tale prototipo avrà il compito di scoprire le varie problematiche che potrebbero presentarsi nel corso del progetto.

#### **5.1.1 Framework AbFmw**

Il framework AbFmw (Abitat Framework) è un insieme di classi sviluppate dall'azienda che si pone come interfaccia tra l'applicazione utente e i servizi offerti da MapGuide o altri tipi di basi di dati, permettendo di rendere indipendente lo sviluppo dell'applicazione dal particolare DBMS utilizzato.

#### **5.1.2 Platform Land® versione base**

Platform Land® è un'applicazione sviluppata precedentemente dall'azienda che, nella sua versione base (senza moduli aggiuntivi), integra già al suo interno il visualizzatore di mappe di Mapguide con le funzionalità connesse e permette di effettuare ricerche sui dati cartografici inseriti, report basilari sugli oggetti selezionati e stampe degli elementi visualizzati. L'applicazione utilizza una base di dati su PostgreSQL che contiene dati di configurazione per poter essere personalizzata dall'utente.

### **5.2 Piano di progetto**

Per raggiungere gli obiettivi previsti ho pianificato di eseguire le seguenti attività:

- Formazione sulle applicazioni che si dovranno utilizzare nel corso del progetto
- Analisi dei requisiti
- Progettazione a partire dalle funzionalità su cui si basa l'intera applicazione come l'inserimento dei clienti e le ricerche, tralasciandone altre che sono già realizzate da Platform Land® come la stampa.
- Sviluppo e Test
- Analisi dei risultati raggiunti e possibili miglioramenti

## **5.3 Formazione**

All'inizio del tirocinio è stato svolto un periodo di formazione che ha permesso di acquisire una certa dimestichezza con le varie applicazioni.

### **5.3.1 OSGeo MapGuide**

Per quanto riguarda MapGuide è stata utilizzata l'applicazione d'esempio (Viewer Sample Application) per esercitarsi con MapGuide Maestro e MapGuide Studio e capire come si possono configurare le fonti dati, i layer, i layout e le mappe.

### **5.3.2 Platform Land®**

Successivamente si è utilizzato Platform Land® per comprendere come era strutturato il software, come funzionavano le ricerche e per conoscere le classi del framework AbFmw. Nel corso di questo periodo è stata anche modificata la grafica per adattarla al prodotto finale.

## **5.4 Analisi dei Requisiti**

### **5.4.1 Casi d'uso**

Durante l'analisi dei requisiti sono stati realizzati dei casi d'uso per le funzionalità che non sono già state realizzate per Platform Land® allo scopo di facilitare l'identificazione dei requisiti funzionali dell'applicazione.

#### ***Inserimento dei clienti in mappa***

1. L'utente richiede l'inserimento dei clienti in mappa
2. L'applicazione richiede i dati dei clienti contenuti nel CRM aziendale
3. L'applicazione per ogni cliente effettua la geocodifica dell'indirizzo
4. L'applicazione per ogni cliente ne memorizza i dati e il punto geografico ottenuto
5. L'applicazione avvisa che l'operazione è terminata

Estensioni:

1a: I dati sono già stati inseriti.

1. Il sistema avvisa l'utente che l'operazione non può essere eseguita
2. L'utente può aggiornare i dati inseriti o cancellarli

3a: Non è stato possibile geocodificare l'indirizzo di un cliente

1. Il sistema memorizza temporaneamente i dati del cliente non geocodificato.

2. Il sistema alla fine dell'elaborazione genera un report con i dati dei clienti non inseriti.

#### **Ricerca in base a parametri**

1. L'utente inserisce i parametri secondo i quali effettuare la ricerca.
2. Il sistema evidenzia il risultato trovato
3. Il sistema effettua uno zoom sul risultato trovato

Estensioni:

1a: L'utente potrebbe ricordare solo in parte i parametri da inserire.

1. Il sistema deve essere in grado di fornire un autocompleter per ogni dato di cui è richiesto l'inserimento
2. L'utente sceglie l'opzione più corretta

2a: L'utente potrebbe non ricordare i parametri da inserire

1. Il sistema deve essere in grado di fornire una lista per ogni dato da inserire contenente tutte le opzioni disponibili.
2. L'utente sceglie l'opzione più corretta

### **5.4.2 Tabella dei requisiti**

#### **5.4.2.1 Requisiti funzionali**

R1	Letture dei dati relativi all'indirizzo dei clienti contenuti nel CRM
R2	Geocodifica degli indirizzi ottenuti in un punto di coordinate (x,y)
R3	Visualizzazione di report sui clienti non geocodificati
R4	Cancellazione dei dati geocodificati
R5	Aggiornamento dei dati geocodificati
R6	Salvataggio del punto geometrico associato al cliente su Shape
R7	Interfaccia per l'inserimento di provincia e comune
R8	Scelta della provincia attraverso un autocompleter
R9	Scelta della provincia attraverso una lista
R10	Scelta del comune attraverso un autocompleter
R11	Scelta del comune attraverso una lista
R12	Zoom sul comune trovato
R13	Evidenziazione del comune trovato
R14	Interfaccia per l'inserimento di provincia, comune e indirizzo
R15	Scelta dell'indirizzo attraverso un autocompleter
R16	Scelta dell'indirizzo attraverso una lista
R17	Zoom sulle tratte associate all'indirizzo cercato

R18	Evidenziazione delle tratte associate all'indirizzo cercato
R19	Interfaccia per l'inserimento di provincia, comune, indirizzo e civico
R20	Zoom sulla tratta in cui si trova il civico cercato
R21	Interfaccia per l'inserimento del nominativo del cliente
R22	Scelta del nominativo attraverso un autocompleter
R23	Scelta del nominativo attraverso una lista
R24	Zoom sul cliente cercato
R25	Evidenziazione del cliente cercato
R26	Salvataggio dei dati associati ai clienti su Shape
R27	Anteprima di stampa
R28	Stampa di una porzione di mappa

#### 5.4.2.2 *Requisiti di vincolo*

R29	Utilizzare il più possibile le classi definite nel framework AbFmw
-----	--

#### 5.4.2.3 *Requisiti di qualità*

R30	Indipendenza dalla specifica piattaforma che fornisce i dati cartografici
R31	Indipendenza dalla piattaforma CRM utilizzata

#### 5.4.3 **Tabella di tracciamento**

Requisito utente	Requisito software	Fonte
Inserimento dei clienti in mappa	R1	Capitolato
	R2	
	R3	
	R4	
	R5	
	R6	
	R29	
	R30	
R31		
Visualizzazione delle geometrie relative ai clienti	R6	Capitolato
Visualizzazione dei dati dei clienti	R26	Capitolato
Ricerca per comune	R7	Capitolato
	R8	
	R9	
	R10	
	R11	
	R12	
	R13	

	R29	
Ricerca per comune, indirizzo	R14 R15 R16 R17 R18 R29	Capitolato
Ricerca per comune, indirizzo, civico	R19 R20 R29	Capitolato
Ricerca per cliente	R21 R22 R23 R24 R25 R29	Capitolato
Filtri statici sui dati dei clienti (settore, tipo, utente assegnato e categoria)	R26	Capitolato
Stampa	R28 R29	Capitolato

## 5.5 Progettazione

### 5.5.1 Definizione del prodotto

Il prodotto andrà ad integrarsi all'applicazione Platform Land® completandola con le funzionalità mancanti e apportando modifiche dove richiesto (ad esempio la realizzazione delle ricerche). Attraverso le classi definite nel framework AbFmw, interagirà con i DBMS richiesti (MySQL per il database del CRM e PostgreSQL per il database della piattaforma Tele Atlas Multinet) e con i servizi messi a disposizione da OSGeo MapGuide.

### 5.5.2 Descrizione dei componenti

#### 5.5.2.1 Platform Land®

Platform Land® utilizza due file di configurazione:

- psw.php che contiene i parametri per effettuare le connessioni ai DBMS utilizzati
- siteconstants.php che contiene le costanti utilizzate dall'applicazione



Il primo deve essere modificato aggiungendo i parametri per la connessione al DBMS del CRM, mentre il secondo deve essere modificato aggiungendo la variabile \$crm che identifica il crm con il quale si vuole interagire e la variabile \$geocoder che definisce la piattaforma utilizzata per effettuare la geocodifica.

#### **5.5.2.2 Framework AbFmw**

Per quanto riguarda il framework si utilizzeranno le seguenti classi:

- DbHanDBMS per potersi connettere ai vari DBMS ed effettuare le query necessarie
- DbHanFDO per potersi connettere al database cartografico di Tele Atlas Multinet
- DrawingExtension per poter memorizzare i dati dei clienti e i relativi punti ottenuti dalla geocodifica nella tabella aziende in formato shape.
- AbFmw per poter ottenere le coordinate della tratta in cui si trova il civico durante l'operazione di geocodifica

#### **5.5.2.3 Tele Atlas Multinet**

Della base di dati fornita da Tele Atlas verranno utilizzate due parti: una su database PostgreSQL costituita dalle tabelle ita\_a7, che contiene le informazioni delle province, e ita\_gc, utilizzata per l'operazione di geocodifica e una su database cartografico in formato sdf costituita dalle tabelle ita\_a7 per le province, ita\_a8 per i comuni, ita\_bu che contiene le informazioni relative alle aree in cui sono presenti edifici, ita\_rr per la rete ferroviaria e ita\_nw per la rete stradale. Per ognuna delle tabelle della seconda parte verrà creato un layer per comporre la mappa di base.

#### **5.5.2.4 Componente per la lettura dei dati dal CRM**

Per rendere l'applicazione indipendente dal software di CRM utilizzato è stata progettata la classe astratta ReaderCRM che implementa il design pattern Abstract Factory e definisce l'interfaccia messa a disposizione per interagire con il CRM. La classe ReaderCRMAbitat è la classe che si occupa della restituzione dei dati contenuti nel CRM dell'azienda mettendo a disposizione il metodo GetAziende() che restituisce un array contenente i dati dei clienti.

#### **5.5.2.5 Componente per l'operazione di geocodifica**

Per rendere l'applicazione il più possibile indipendente dalla piattaforma utilizzata per la geocodifica è stata progettata l'interfaccia GeocoderInt. Il componente che implementa tale interfaccia è la classe GeocoderTA, realizzando l'operazione di geocodifica basandosi sui dati cartografici forniti da Tele Atlas. In questo caso non è stato possibile utilizzare il design pattern Abstract Factory, come inizialmente si era

pensato, a causa delle diversità tra le piattaforme dalle quali è possibile ricavare l'operazione di geocodifica.

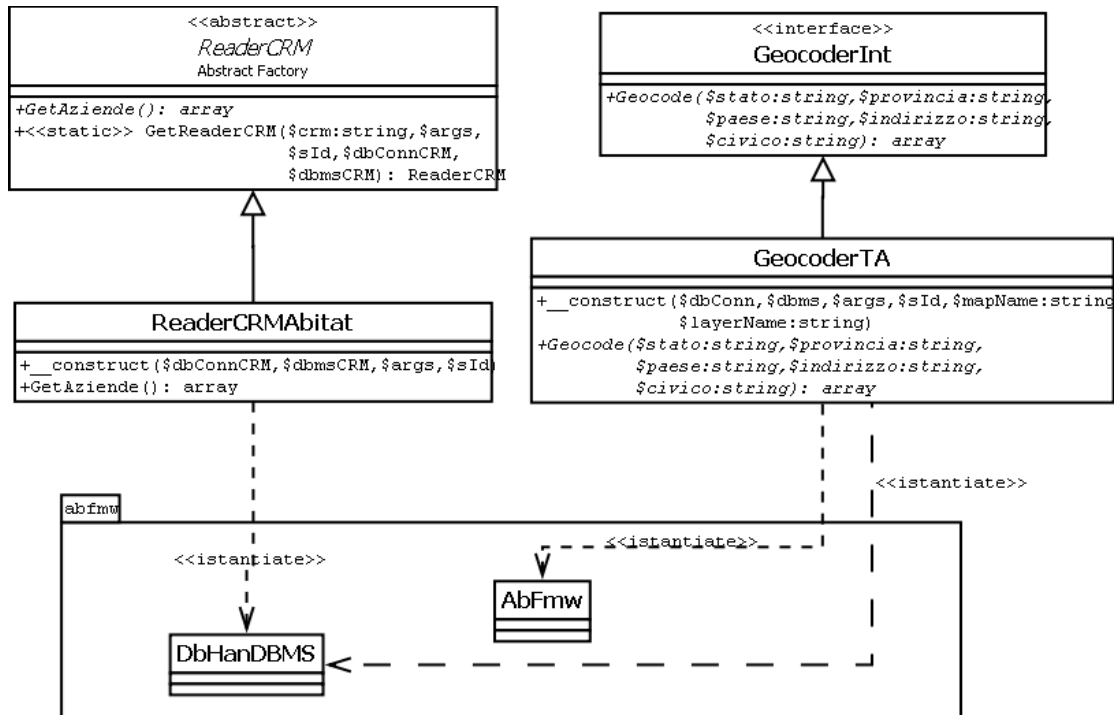


FIGURA 3: SCHEMA UML DELLE CLASSI PROGETTATE

### 5.5.2.6 Componente per l'autocompleter

Come per le ricerche messe a disposizione da Platform Land®, è stata utilizzata un'applicazione già pronta contenuta nella libreria Javascript Scriptaculous che permette di usare la funzione di autocompletamento.

### 5.5.3 Diagrammi di attività

Data la complessità dell'algorithm necessario per la geocodifica ho creato un diagramma di attività in modo tale da renderne più facile la realizzazione.

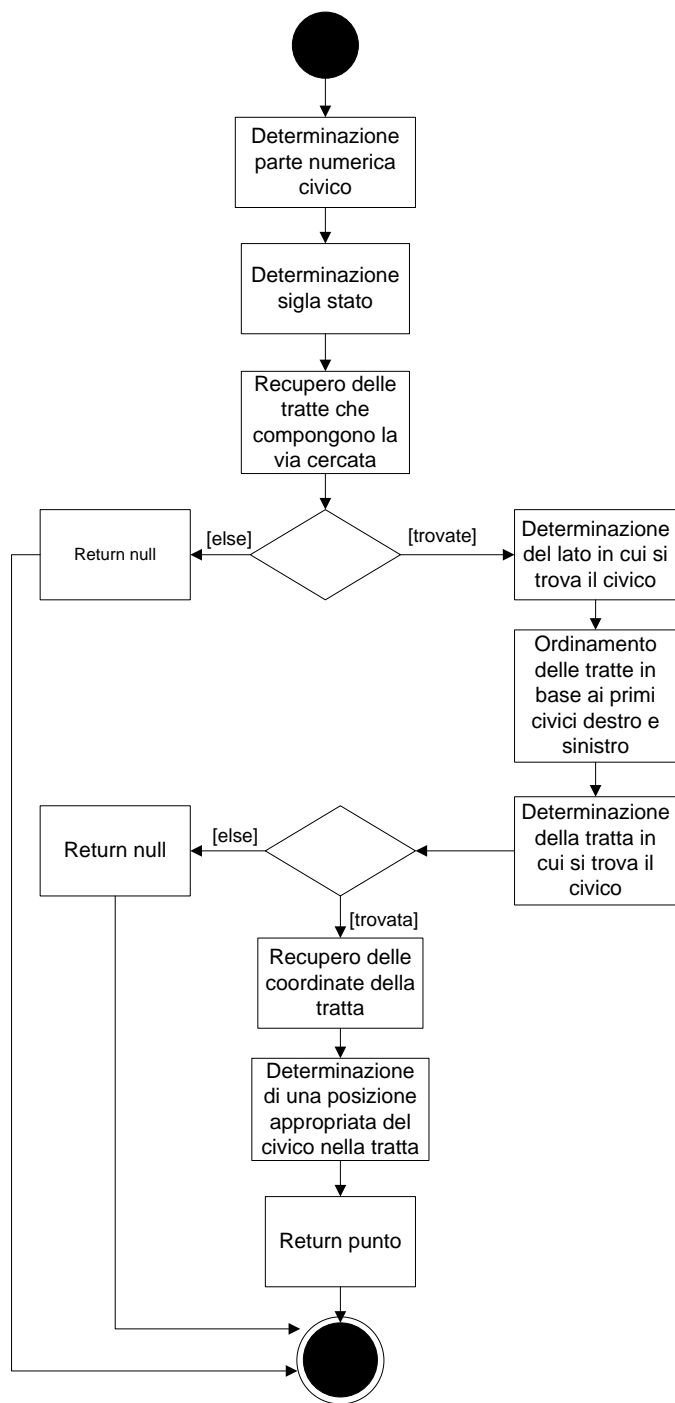


FIGURA 4: DIAGRAMMA DI ATTIVITÀ PER L'ALGORITMO DI GEOCODIFICA

#### 5.5.4 Design Pattern

Nella progettazione del componente associato alla lettura dei dati è stato utilizzato il design pattern Abstract Factory. Tale pattern ha come scopo il fornire un'interfaccia per creare famiglie di oggetti dipendenti tra loro senza dover specificare il nome delle classi concrete.

#### 5.5.5 Tracciamento requisiti – componenti

Requisito software	Componente
R1	Classe ReaderCRMabitat
R2	Classe GeocoderTA
R3	caricaAziende.php
R4	funzioniAziende.php
R5	funzioniAziende.php
R6	caricaAziende.php
R7	ricerca_comune.php
R8	Libreria Scriptaculous ricerca_comune_provincia_action.php
R9	ricerca_comune_provincia_getList.php
R10	Libreria Scriptaculous ricerca_comune_comune_action.php
R11	ricerca_comune_comune_getList.php
R12	Platform Land®
R13	highlight_ricerca.js
R14	ricerca_comune_indirizzo.php
R15	Libreria Scriptaculous ricerca_comune_indirizzo_indirizzo_action.php
R16	ricerca_comune_indirizzi_indirizzo_getList.php
R17	Platform Land®
R18	highlight_ricerca.js
R19	ricerca_comune_indirizzo_civico.php
R20	Platform Land®
R21	ricerca_cliente.php
R22	Libreria Scriptaculous ricerca_cliente_action.php
R23	ricerca_cliente_getList.php
R24	Platform Land®
R25	highlight_ricerca.js
R26	caricaAziende.php
R27	Platform Land®
R28	Platform Land®
R29	Intero prodotto

R30	Interfaccia GeocoderInt
R31	Classe astratta ReaderCRM

### 5.5.6 Tracciamento requisiti – Metodi

Requisito software	Metodo
R1	Classe ReaderCRMabitat::GetAziende()
R2	Classe GeocoderTA::Geocode()
R3	caricaAziende.php
R4	funzioniAziende.php
R5	funzioniAziende.php
R6	caricaAziende.php::insertPuntoAzienda()
R7	ricerca_comune.php
R8	Libreria Scriptaculous::Ajax.AutoCompleteer() ricerca_comune_provincia_action.php
R9	ricerca_comune_provincia_getList.php
R10	Libreria Scriptaculous::Ajax.AutoCompleteer() ricerca_comune_comune_action.php
R11	ricerca_comune_comune_getList.php
R12	Platform Land®
R13	highlight_ricerca.js::Highlight()
R14	ricerca_comune_indirizzo.php
R15	Libreria Scriptaculous::Ajax.AutoCompleteer() ricerca_comune_indirizzo_indirizzo_action.php
R16	ricerca_comune_indirizzi_indirizzo_getList.php
R17	Platform Land®
R18	highlight_ricerca.js::Highlight()
R19	ricerca_comune_indirizzo_civico.php
R20	Platform Land®
R21	ricerca_cliente.php
R22	Libreria Scriptaculous::Ajax.AutoCompleteer() ricerca_cliente_action.php
R23	ricerca_cliente_getList.php
R24	Platform Land®
R25	highlight_ricerca.js::Highlight()
R26	caricaAziende.php::insertPuntoAzienda()
R27	Platform Land®
R28	Platform Land®
R29	Intero prodotto
R30	Interfaccia GeocoderInt
R31	Classe astratta ReaderCRM::GetReaderCRM()

## 5.6 Realizzazione e Collaudo

La realizzazione è consistita nello sviluppo della funzionalità di inserimento dei clienti in mappa e delle varie ricerche oltre ai filtri statici per i quali è bastato configurare dei layer utilizzando MapGuide Maestro. Tali layer si basano sullo shape contenente i dati dei clienti memorizzati alla fine della procedura di inserimento. Per effettuare i test sono stati utilizzati il debugger incluso in Eclipse per testare il codice PHP e il plug-in Firebug di Firefox che, oltre a permettere di testare il codice Javascript, consente di misurare la durata delle richieste effettuate attraverso AJAX.

### 5.6.1 Inserimento dei clienti in mappa

Per quanto riguarda l'operazione di inserimento non sono state sviluppate le funzioni di aggiornamento e cancellazione, ma solamente la funzione di caricamento dei dati.

#### 5.6.1.1 *funzioniAziende.php*

```
<?php
session_start();
require_once '../include/siteconstants.php';

$args= ($_SERVER['REQUEST_METHOD']=='POST') ? $_POST : $_GET;
$comando = '';
if(array_key_exists('comando',$args)){
    $comando = $args['comando'];
}
?>
<html>
<head>
    <title><?php echo $title; ?></title>
    <link href="../Stili.css" rel="stylesheet" type="text/css">

    <script type="text/javascript" src="../<?php echo $frameworkPath
?>js/prototype.js"></script>
    <script type="text/javascript" src="../<?php echo $frameworkPath
?>js/effects.js"></script>
    <script type="text/javascript" src="../<?php echo $frameworkPath
?>js/controls.js"></script>
    <script language="javascript"
src="../../../../viewerfiles/browserdetect.js"></script>
    <script language="javascript" src="../<?php echo $frameworkPath
?>js/json.js"></script>

    <script language = "JavaScript" type = "text/JavaScript">
```

```

function changeCommand(comando){
    document.getElementById('comando').value = comando;
    document.getElementById('form').submit();
}
</script>

</head>

<body style="padding:0; margin: 0; background: white; background-image:
url('.././moduli/modulo_base/image/dimming/pf_report_bottom.jpg');
background-repeat: repeat-x; background-position: 0px 150px;">
<form style="margin: 0px; margin-top:30px; padding:0px;" id="form"
action="funzioniAziende.php" target="" method="POST" onKeyPress="">
    <input type="hidden" name="comando" id="comando" value="<?php echo
$comando?>" style="display:none; width:0px"/>
    <table>
        <tr>
            <td width="350">
                <label class="reportStile1">Vuoi davvero caricare i dati?</label>
            </td>
        </tr>
        <tr align="center">
            <td style="padding-bottom: 10px;">
                <input type="button" value="Carica"
onclick="changeCommand('CARICA')" class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
                <input type="button" value="Aggiorna" onclick=""
class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
                <input type="button" value="Elimina" onclick=""
class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
                <input type="button" value="Chiudi" onclick="parent.nascondi()"
class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
            </td>
        </tr>
        <tr>
            <td height="100" valign="top">

```

```

        <div id="image" style="display:none" align="center">
            
        </div>
    </td>
</tr>
</table>
</form>
<script language = "JavaScript" type = "text/JavaScript">
    var comando = '<?php echo $comando?>';
    var messaggio = '<?php echo $messaggio?>';

    switch(comando){
    case 'CARICA':
        if (msie)
            reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
        else
            reqHandler = new XMLHttpRequest();

        reqHandler.open("POST", "caricaAziende.php", true);
        reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
        reqHandler.onreadystatechange = function(){
            if (reqHandler.readyState == 4 && reqHandler.status == 200) {
                document.getElementById('image').style.display='none';
                alert(reqHandler.responseText);
            }
        };
        document.getElementById('image').style.display='';
        reqHandler.send(null);
        break;
    case 'AGGIORNA':
        <?php //TODO:gestire comando aggiorna?>
    case 'ELIMINA':
        <?php //TODO:gestire comando elimina?>
    }
    </script>
</body>
</html>

```

### 5.6.1.2 *caricaAziende.php*

```

<?php
session_start();
require_once '../include/siteconstants.php';
require_once 'ReaderCRM.php';
require_once 'GeocoderTA.php';
require_once '../'.$frameworkPath.'DrawingExtension.php';

```



```

require_once '../'.$frameworkPath.'DbHanFDO.php';

MgInitializeWebTier($configFilePath);
$json = new Services_JSON();
$responseType = '';
$response = '';
$args= ($_SERVER['REQUEST_METHOD']=='POST') ? $_POST : $_GET;
$sId = $_SESSION['MgSessionId'];
$mapName = 'Map';

$dbHan = new DbHanFDO($args, $_SESSION['MgSessionId'],'Map');
$aziendeCaricate = $dbHan->Select('FeatId', 'AZIENDE', "", 1);
unset($dbHan);
if($aziendeCaricate[0][0]==null){
    //acquisizione aziende
    $dbConnCRM = connectTo('crm', $dbConnSet);
    $reader =
ReaderCRM::GetReaderCRM($crm,$args,$sId,$dbConnCRM,$dbConnSet['crm']['DBMS']
);
    $aziende = $reader->GetAziende();
    unset($reader);
    $dbConnCRM -> Close();
    //acquisizione geocoder giusto
    $dbConn = null;
    $geo = null;
    switch($geocoder){
        case 'Teleatlas':
            $dbConn = connectTo('mgos',$dbConnSet);
            $geo = new
GeocoderTA($dbConn,$dbConnSet['mgos']['DBMS'],$args,$sId,$mapName,'RETE');
            break;
            //aggiungere qui i case per gestire google e bing
    }
    $length = count($aziende);
    $aziendeNotInserted = array(); //array aziende non inserite
    $numAziendeInserted = 0; //numero aziende inserite
    //geocodifica e inserimento per ogni azienda
    for($i=0; $i<$length; $i++){
        $lengthRow = count($aziende[$i]);

        $stato = $aziende[$i][$lengthRow - 5];
        $provincia = $aziende[$i][$lengthRow - 4];
        $paese = $aziende[$i][$lengthRow - 3];
        $indirizzo = $aziende[$i][$lengthRow - 2];
        $civico = $aziende[$i][$lengthRow - 1];

        $punto = $geo->Geocode($stato,$provincia,$paese,$indirizzo,$civico);

```

```

        if ($punto == null){
            $aziendeNotInserted[count($aziendeNotInserted)] = $aziende[$i];
        }
        else{
            insertPuntoAzienda($punto,$aziende[$i]);
            $numAziendeInserted++;
        }
    }
    if($dbConn != null){
        $dbConn -> Close();
    }
    //operazioni da svolgere per attivare il report e per creare il file di
log
    $responseType = 'text/plain';
    $response = $json->encode("Caricamento completato.");
}
else{
    $responseType = 'text/plain';
    $response = $json->encode("Dati caricati in precedenza.");
}
header('Content-Type: ' . $responseType);
echo $response;

function insertPuntoAzienda($punto,$azienda){
    //TODO:gestire major e istat
    global $sId;
    global $args;
    global $mapName;

    $draw = new DrawingExtension($args, $sId, $mapName);
    $layerName = "AZIENDE";
    $geometryType = "point";
    $xs[0] = $punto["x"];
    $ys[0] = $punto["y"];
    $nomeCampiStr = "COD_AZIENDA,SETTORE,TIPO,ASSEGNATO_A,CAT_NOMINAT";
    $valoriStr =
$azienda[0].",".$azienda[1].",".$azienda[2].",".$azienda[3].",".$azienda[4];

    $puntoInserted = $draw -> DrawFeature($layerName, $geometryType, $xs, $ys,
$nomeCampiStr, $valoriStr, null,0,0);
}
?>

```

### **5.6.1.3 ReaderCRM.php**

```

<?php
/*

```

```

* AUTORE: Claudio Gecchele
* E-MAIL claudio.gecchele@abitat.it
* DATA CREAZIONE: 05/07/2010
*/
require_once 'ReaderCRMabitat.php';

abstract class ReaderCRM{

    //fornisce una matrice contenente i dati delle aziende
    public abstract function GetAziende();

    public static function GetReaderCRM($crm, $args, $sId, $dbConnCRM, $dbmsCRM) {
        //controllo parametri definiti
        if(is_null($crm) || !isset($crm)){
            throw new UndefinedParamException("Parametro crm non definito.");
        }
        if(is_null($args) || !isset($args)){
            throw new UndefinedParamException("Parametro args non definito.");
        }
        if(is_null($sId) || !isset($sId)){
            throw new UndefinedParamException("Parametro sId non definito.");
        }
        if(is_null($dbConnCRM) || !isset($dbConnCRM)){
            throw new UndefinedParamException("Parametro dbConn non definito.");
        }
        if(is_null($dbmsCRM) || !isset($dbmsCRM)){
            throw new UndefinedParamException("Parametro dbms non definito.");
        }
        //scelta del reader giusto
        switch($crm){
            case 'Abitat':return new
ReaderCRMabitat ($dbConnCRM, $dbmsCRM, $args, $sId);
        }
    }
}

```

#### **5.6.1.4 ReaderCRMabitat.php**

```

<?php
/*
* AUTORE: Claudio Gecchele
* E-MAIL claudio.gecchele@abitat.it
* DATA CREAZIONE: 05/07/2010
*/
require_once '../'.$frameworkPath.'DbHanDBMS.php';

class ReaderCRMabitat extends ReaderCRM{
    private $args;

```

```

private $sId;
private $dbConnCRM;
private $dbmsCRM;

public function __construct($dbConnCRM,$dbmsCRM,$args,$sId){
    //controllo parametri definiti
    if(is_null($args) || !isset($args)){
        throw new UndefinedParamException("Parametro args non definito.");
    }
    if(is_null($sId) || !isset($sId)){
        throw new UndefinedParamException("Parametro sId non definito.");
    }
    if(is_null($dbConnCRM) || !isset($dbConnCRM)){
        throw new UndefinedParamException("Parametro dbConn non definito.");
    }
    if(is_null($dbmsCRM) || !isset($dbmsCRM)){
        throw new UndefinedParamException("Parametro dbms non definito.");
    }
    $this->args = $args;
    $this->sId = $sId;
    $this->dbConnCRM = $dbConnCRM;
    $this->dbmsCRM = $dbmsCRM;
}

/*restituisce una matrice in cui ogni riga contiene i dati di una azienda
* tenendo i dati geografici per ultimi:
*[row][0]=id_azienza
*[row][1]=settore
*[row][2]=tipo
*[row][3]=assegnatoA
*[row][4]=cat_nominativi
*[row][5]=major
*[row][6]=istat
*[row][7]=stato
*[row][8]=provincia
*[row][9]=paese
*[row][10]=indirizzo
*[row][11]=civico
*/
public function GetAziende(){

    $dbHan = new DbHanDBMS($this->dbConnCRM,$this->dbmsCRM,$this->args,$this->sId);

    //TODO: mancano major e istat in quanto non sono presenti nel CRM locale.

```

```

    $nomeCampiStr =
"ACCOUNTS.ID,INDUSTRY,ACCOUNT_TYPE,USER_NAME,CATEGORIA_NOMINATIVI_C,BILLING_
ADDRESS_COUNTRY,BILLING_ADDRESS_STATE,BILLING_ADDRESS_CITY,BILLING_ADDRESS_S
TREET";
    $nomeTabelleStr = "ACCOUNTS,ACCOUNTS_CSTM,USERS";
    $where = "(ASSIGNED_USER_ID=USERS.ID) AND (ACCOUNTS.ID=ID_C)";
    $response = $dbHan->Select($nomeCampiStr,$nomeTabelleStr,$where,1);
    unset($dbHan);

    $length = count($response);
    for($i = 0; $i <$length; $i++){
        $lengthRow = count($response[$i]);           //lunghezza riga

        //divido l'indirizzo in nome della via e numero civico
        $address = explode(",",$response[$i][$lengthRow-1]);

        //elimino gli spazi dal numero civico e lo aggiungo alla risposta
        $response[$i][$lengthRow] = trim($address[1]);

        //aggiorno l'indirizzo
        $response[$i][$lengthRow-1] = $address[0];
    }
    return $response;
}
}
}

```

### **5.6.1.5 GeocoderInt.php**

```

<?php
/*
 * AUTORE: Claudio Gecchele
 * E-MAIL claudio.gecchele@abitat.it
 * DATA CREAZIONE: 18/08/2010
 */
interface GeocoderInt{

    //restituisce latitudine e longitudine associate ai parametri
    //restituisce null nel caso in cui non riesca a geocodificare i parametri
    public function geocode($stato,$provincia,$paese,$indirizzo,$civico);
}
?>

```

### **5.6.1.6 GeocoderTA.php**

```

<?php
/*
 * AUTORE: Claudio Gecchele
 * E-MAIL claudio.gecchele@abitat.it
 * DATA CREAZIONE: 18/08/2010

```

```

*/
require_once '../'.$frameworkPath.'DbHanDBMS.php';
require_once 'GeocoderInt.php';
require_once '../'.$frameworkPath.'AbFmw.php';

class GeocoderTA implements GeocoderInt{
    private $args;
    private $sId;
    private $dbConn;
    private $dbms;
    private $mapName;
    private $layerName;

    public function __construct($dbConn,$dbms,$args,$sId,$mapName,$layerName){
        //controllo parametri definiti
        if(is_null($args) || !isset($args)){
            throw new UndefinedParamException("Parametro args non definito.");
        }
        if(is_null($sId) || !isset($sId)){
            throw new UndefinedParamException("Parametro sId non definito.");
        }
        if(is_null($dbConn) || !isset($dbConn)){
            throw new UndefinedParamException("Parametro dbConn non definito.");
        }
        if(is_null($dbms) || !isset($dbms)){
            throw new UndefinedParamException("Parametro dbms non definito.");
        }
        if(is_null($mapName) || !isset($mapName)){
            throw new UndefinedParamException("Parametro mapName non definito.");
        }
        if(is_null($layerName) || !isset($layerName)){
            throw new UndefinedParamException("Parametro layerName non
definito.");
        }
        $this->args = $args;
        $this->sId = $sId;
        $this->dbConn = $dbConn;
        $this->dbms = $dbms;
        $this->mapName = $mapName;
        $this->layerName = $layerName;
    }

    //restituisce latitudine e longitudine associate ai parametri
    //restituisce null nel caso in cui non riesca a geocodificare i parametri
    public function Geocode($stato,$provincia,$paese,$indirizzo,$civico){

```

```

//determinazione parte numerica del civico
$numCivico = "";
for($i = 0; $i<strlen($civico); $i++){
    $char = substr($civico,$i,1);
    if(($char >0)&&($char <= 9)){
        $numCivico .= $char;
    }
}

$siglaStato = strtoupper(substr($stato,0,3));

/*recupero featID,primo civico sinistro,ultimo civico sinistro, primo
civico destro e ultimo civico destro dei tronchetti che compongono la
via cercata
senza controllare nella tabella ih(intermediate)
occorre tenerne conto nel caso nella tabella gc L_INTM o R_INTM sono
uguali a 1
*/
$dbHan = new DbHanDBMS($this->dbConn, $this->dbms, $this->args, $this->
sId);
$nomeCampiStr = "ID,L_F_ADD,L_T_ADD,R_F_ADD,R_T_ADD";
$nomeTabelleStr = "ita_gc";
$where = "((L_LAXONLC='" . $siglaStato . "')OR(R_LAXONLC='".
$siglaStato ."''))AND";
$where .= "((L_AXON='" . $provincia . "')OR(R_AXON='". $provincia
.''))AND";
$where .= "((L_LAXON='" . $paese . "')OR(R_LAXON='". $paese ."''))AND";
$where .= "(FULLNAME='" . $indirizzo ."' )AND";
$where .= " NOT(L_F_ADD=-1 AND R_F_ADD=-1)";
$response = $dbHan-> Select($nomeCampiStr, $nomeTabelleStr,$where);
unset($dbHan);

if($response != false){
    /*sol: side of line ovvero sinistra 1 destra 3
    1 è l'indice del primo civico a sinistra
    3 è l'indice del primo civico a destra
    cerco in tutti i tronchetti restituiti perchè alcuni attributi
    hanno -1 come valore del civico
    */
    $sol = null;
    $length = count($response);
    for($i = 0; $i < $length && $sol == null; $i++){
        if($response[$i][1] != -1 && $response[$i][1]%2 == $numCivico%2){
            $sol = 1;
        }else if($response[$i][3] != -1 && $response[$i][3]%2 ==
$numCivico%2){
            $sol = 3;

```

```

    }
}
/*prima devo ordinare i civici
sol indica il campo secondo cui ordinare
uso anche l'altro valore di sol in quanto, probabilmente a causa
di errori nei dati teleatlas, esistono tronchetti diversi che iniziano
con lo stesso civico
*/
$response = $this->bubbleSort($response,$sol,($sol==1) ? 3 : 1);

$featId;
$primoCivico;
$ultimoCivico;
$trovato = false;
if($sol != null){
    for($i = 0; $i < $length && !$trovato; $i++){
        $minCivico;
        $maxCivico;
        /*determino ordine dei civici
        cioè se il civico più piccolo è il primo o l'ultimo
        della parte della strada in cui sto cercando.
        Viene eseguito ogni volta a causa dei possibili civici che valgono
-1
        */
        if ($response[$i][$sol] <= $response[$i][$sol + 1]){
            $minCivico = $response[$i][$sol];
            $maxCivico = $response[$i][$sol + 1];
        }
        else{
            $minCivico = $response[$i][$sol + 1];
            $maxCivico = $response[$i][$sol];
        }

        /* determino il featId, il primo e l'ultimo civico
        del tronchetto in cui si trova il civico cercato
        Se il civico non si trova nei dati teleatlas viene messo
        dove si trova il primo civico o dove si trova l'ultimo civico
        a seconda se il civico cercato è inferiore o superiore a
        quelli presenti sui dati Teleatlas
        */
        if($numCivico < $minCivico){
            //se il civico cercato non è compreso nei dati teleatlas
            //lo piazza dove si trova il primo civico
            $featId = $response[$i][0];
            $primoCivico = $response[$i][$sol];
            $ultimoCivico = $response[$i][$sol + 1];
            $numCivico = $primoCivico;

```



```

        $trovato = true;
    }else if($numCivico >= $minCivico && $numCivico <= $maxCivico ){
        //il civico è compreso recupero solo i dati
        $featId = $response[$i][0];
        $primoCivico = $response[$i][$sol];
        $ultimoCivico = $response[$i][$sol + 1];
        $trovato = true;
    }else if(($i + 1) == $length){
        //se il civico è oltre i dati teleatlas lo piazzò dove si
        //trova l'ultimo civico
        $featId = $response[$i][0];
        $primoCivico = $response[$i][$sol];
        $ultimoCivico = $response[$i][$sol + 1];
        $numCivico = $ultimoCivico;
        $trovato = true;
    }
}
}

if($trovato){
    //recupero coordinate tronchetto
    $abfmw = new AbFmw($this->args,$this->id);
    $filter = "(ID=" . $featId . ")";
    $tronchetto = $abfmw -> GetCoordinateByLayerName($this->layerName,
$this->mapName, $filter);
    $xs = $tronchetto[0]["xs"];
    $ys = $tronchetto[0]["ys"];
    $punto = array();

    //determinazione numero dei civici sul lato della strada
    //potrebbe accadere che i civici dispari e pari si sormontino
    //in quanto vengono posizionati sulla strada, non a lato
    $numCivici = abs( $ultimoCivico - $primoCivico)/2 + 1;
    //determinazione posizione civico
    $posCivico = abs( $numCivico - $primoCivico)/2 + 1;
    /*se c'è un solo civico lo piazzò a metà tronchetto
    utilizzando le coordinate più centrali possibile
    */
    if($numCivici == 1){
        $punto["x"] = ($xs[ floor( (count($xs) -1) /2) ] + $xs[ floor(
(count($xs) -1) /2) + 1])/2;
        $punto["y"] = ($ys[ floor( (count($ys) -1) /2) ] + $ys[ floor(
(count($ys) -1) /2) + 1])/2;
    }
    // se è il primo lo piazzò tra il primo e il secondo punto
    else if($posCivico == 1){
        $punto["x"] = ($xs[1] + $xs[0])/2;
    }
}
}
}

```

```

    $punto["y"] = ($ys[1] + $ys[0])/2;
}
//se è l'ultimo lo piazzo tra l'ultimo e il penultimo punto
else if($posCivico == $numCivici){
    $punto["x"] = ($xs[count($xs) - 1] + $xs[count($xs) - 2])/2;
    $punto["y"] = ($ys[count($ys) - 1] + $ys[count($ys) - 2])/2;
}
//se è in mezzo trovo una posizione appropriata
else{
    $posizionato = false;
    do{
        $numPunti = count($xs);
        $puntiAlCivico = $numPunti/$numCivici;
        //se ci sono abbastanza punti per civico
        if($puntiAlCivico >= 1){
            $index = floor(($posCivico -1) * $puntiAlCivico);
            $punto["x"] = $xs[$index];
            $punto["y"] = $ys[$index];
            $posizionato = true;
        }
        //altrimenti ne aggiungo n-1 calcolando
        //quelli che stanno in mezzo agli altri
        else{
            $tempX = array(); //array coordinate x
            $tempY = array(); //array coordinate y
            $pos = 0;
            for( $i = 0; $i < $numPunti; $i++){
                $tempX[$pos] = $xs[$i];
                $tempY[$pos] = $ys[$i];
                $pos++;
                /*se non sono alla fine
                calcolo il punto che sta in mezzo fra i punti
                i e i+1 e lo aggiungo all'array.
                */
                if($i+1 < $numPunti){
                    $tempX[$pos] = ($xs[$i] + $xs[$i+1])/2;
                    $tempY[$pos] = ($ys[$i] + $ys[$i+1])/2;
                    $pos++;
                }
            }
            $xs = $tempX; //nuove coordinate x
            $ys = $tempY; //nuove coordinate y
        }
    }while(!$posizionato);
}
return $punto;
}

```

```

        else{
            return null;
        }
    }
    else{
        return null;
    }
}

//array: l'array bidimensionale da ordinare
//index: l'indice del primo campo secondo cui ordinare
//index2: l'indice del secondo campo secondo cui ordinare
private function bubbleSort($array,$index,$index2){
    $ordinato;
    do{
        $ordinato = true;
        $length = count($array);
        for($i=1; $i<$length; $i++){
            if(($array[$i][$index] < $array[$i-1][$index]) ||
($array[$i][$index] == $array[$i-1][$index] && $array[$i][$index2] <
$array[$i-1][$index2])){
                $temp = $array[$i];
                $array[$i] = $array[$i-1];
                $array[$i-1] = $temp;
                $ordinato = false;
            }
        }
    }while(!$ordinato);
    return $array;
}
}
?>

```

## 5.6.2 Ricerca per comune, per comune e indirizzo, per comune, indirizzo e civico e ricerca per cliente

Le ricerche sono state sviluppate apportando modifiche alle ricerche già a disposizione di Platform Land®. In particolare è stata utilizzata anche la provincia per poter distinguere paesi omonimi e nella ricerca per comune, indirizzo e civico per il campo di input riguardante il civico non sono stati sviluppati la visualizzazione della lista e l'autocomplete in quanto sono stati ritenuti poco utili e si è scelto di effettuare soltanto lo zoom sulla tratta per semplicità. Inoltre è stato creato il file `dbhandbms_controller.php` per poter utilizzare la classe `DbHanDBMS` del framework attraverso richieste JSON.

### 5.6.2.1 `dbhandbms_controller.php`

```

<?php
session_start();

require_once '../include/siteconstants.php';
require_once '../'.$frameworkPath.'DbHanDBMS.php';

$args = ($_SERVER['REQUEST_METHOD'] == 'POST')? $_POST : $_GET;
MgInitializeWebTier($configFilePath);
$dbhanObj = null;
$json = new Services_JSON();
$responseType = '';
$response = '';

if(array_key_exists('COMMAND',$args)){
    switch($args['COMMAND']){
        case 'SELECT':
            $dbConn = connectTo('mgos', $dbConnSet);
            $dbhanObj = new
DbHanDBMS($dbConn,$dbConnSet['mgos']['DBMS'],$args,$args['SESSION']);
            $responseType = 'text/plain';
            $response = $json->encode($dbhanObj->Select());
            unset($dbhanObj);
            $dbConn -> Close();
            break;
    }
}

header('Content-Type: ' . $responseType);
echo $response;

```

### **5.6.2.2 *ricerca\_comune.php***

```

<?php
session_start();
include_once '../include/siteconstants.php';
include_once '../' . $frameworkPath . 'AbFmw.php';
include_once '../' . $frameworkPath . 'DbHanFDO.php';
include_once '../' . $frameworkPath . 'js/commonjs.php';

$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

$provincia = "";
$cod_provincia = $args['valore_provincia'];

$dbhan = new DbHanFDO($args, $_SESSION['MgSessionId'],'Map');
$where = "ORDER07='". $cod_provincia . "'";
$response = $dbhan->Select('NAME', 'PROVINCE', $where, 1);
unset($dbhan);

```

```

$provincia = $response[0][0];

$comune = "";
$cod_comune = $args['valore_comune'];

$dbHan = new DbHanFDO($args, $_SESSION['MgSessionId'],'Map');
$where = "ORDER08='" . $cod_comune . "'";
$response = $dbHan->Select('NAME', 'COMUNI', $where, 1);
unset($dbHan);

$comune = $response[0][0];
?>
<html>
  <head>
    <title><?php $title?></title>
    <link href="../../Stili.css" rel="stylesheet" type="text/css">

    <style>
      .autocomplete
      {
        font-family: Tahoma, Arial, Helvetica, sans-serif;
        font-size: 8pt;
        position:absolute;
        background-color:white;
        border:1px solid #888;
        margin:0;
        padding:0;
      }

      .autocomplete ul
      {
        list-style-type:none;
        margin:0;
        padding:0;
      }

      .autocomplete ul li.selected
      {
        background-color: #FFB;
      }

      .autocomplete ul li
      {
        list-style-type:none;
        display:block;
        margin:0;
        padding:2px;

```

```

        cursor:pointer;
    }
</style>

    <script type="text/javascript" src="../../<?php echo $frameworkPath
?>js/prototype.js"></script>
    <script type="text/javascript" src="../../<?php echo $frameworkPath
?>js/effects.js"></script>
    <script type="text/javascript" src="../../<?php echo $frameworkPath
?>js/controls.js"></script>
    <script language="javascript"
src="../../../../../../viewerfiles/browserdetect.js"></script>
    <script language="javascript" src="../../<?php echo $frameworkPath
?>js/json.js"></script>
    <script language="javascript"
src="../../moduli/modulo_base/js/highlight_ricerca.js"></script>

    <script language = "JavaScript" type = "text/JavaScript">
var session = '<?php echo $_SESSION['MgSessionId'] ?>';
var mapName = 'Map';
var caller = "";
var cod_provincia = "<?php echo $cod_provincia ?>";
var cod_comune = "<?php echo $cod_comune ?>";

var mapFrame =
parent.document.getElementById("indexFrame").contentWindow.document.getEleme
ntById("mgFrame").contentWindow.document.getElementById("mapFrame");
var viewerFrame =
parent.document.getElementById("indexFrame").contentWindow.document.getEleme
ntById("mgFrame");

function getList(element,whereClause)
{
    var list = document.getElementById("listBox");
    //debugger;
    var reqParams = "SESSION=" + encodeURIComponent(session);

    var nomeFile="";
    switch(element){
    case 'valore_provincia':
        nomeFile = "ricerca_comune_provincia_getList.php";
        break;
    case 'valore_comune':
        reqParams += "&WHERECLAUSE=" + encodeURIComponent(whereClause);
        nomeFile = "ricerca_comune_comune_getList.php";
        break;
    }
}

```

```

if (msie)
    reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
else
    reqHandler = new XMLHttpRequest();

reqHandler.open("POST", nomeFile, false);
reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

reqHandler.send(reqParams);

var values = reqHandler.responseText.parseJSON();

for(var i=0; i<values.length; i++)
{
    list_el = null;
    list_el = document.createElement("option");

    list_el.value = values[i][0];
    list_el.innerHTML = values[i][1];

    list.appendChild(list_el);
}
}

function cerca()
{
    //debugger;
    var reqParams = "COMMAND=SELECTFEAUTURES";
    reqParams += "&SESSION=" + encodeURIComponent(session);
    reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
    reqParams += "&LAYERNAME=" + encodeURIComponent("COMUNI");
    reqParams += "&LAYERPROPERTIES=" + encodeURIComponent("ID");
    reqParams += "&WHERECLAUSE=";

    var whereClause = "(ORDER08='" + encodeURIComponent(cod_comune) +
    "')";

    reqParams += encodeURIComponent(whereClause);

    if (msie)
        reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
    else
        reqHandler = new XMLHttpRequest();

```

```

        reqHandler.open("POST",
"..../moduli/modulo_base/fdo/abfmw_controller.php", false);
        reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

        reqHandler.send(reqParams);

//debugger;
var values = reqHandler.responseText.parseJSON();

activateGroup("CONFINI");

reqParams = "COMMAND=GETENVELOPE";
reqParams += "&SESSION=" + encodeURIComponent(session);
reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
reqParams += "&LAYERNAME=COMUNI";
reqParams += "&IDFIELD=ID";
reqParams += "&IDS=" + encodeURIComponent(values[0].ID);

if (msie)
    reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
else
    reqHandler = new XMLHttpRequest();

reqHandler.open("POST",
"..../moduli/modulo_base/fdo/abfmw_controller.php", false);
reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

reqHandler.send(reqParams);

envelope = reqHandler.responseText.parseJSON();

if (envelope != "ENVELOPE IS NULL")
    ZoomSimple_new(envelope, mapFrame);
else
    alert ('Comune non Trovato!');

HighlightAllOff();
Highlight(values[0].ID, "ID", "COMUNI");

// Consente l'apertura del report.
window.open("../report_controller.php?KEYS=''&TABLES=COMUNI&SESSION="
+ session + "&clausolaWhere=" +
encodeURIComponent(whereClause), "report1Frame");
}

```



```

function activateGroup(name)
{
    var reqParams = "COMMAND=ACTIVATEGROUP";
    reqParams += "&SESSION=" + encodeURIComponent(session);
    reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
    reqParams += "&LAYERGROUPNAME=" + encodeURIComponent(name);

    if (msie)
        reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
    else
        reqHandler = new XMLHttpRequest();

    reqHandler.open("POST",
"..../moduli/modulo_base/fdo/abfmw_controller.php", false);
    reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

    reqHandler.send(reqParams);

    viewerFrame.contentWindow.Refresh();
}

function changeValue(codice)
{
    document.getElementById(caller).value = codice;
    document.getElementById("form").submit();
}
</script>
</head>
<body style="padding:0; margin: 0; background: white; background-image:
url('../moduli/modulo_base/image/dimming/pf_report_bottom.jpg');
background-repeat: repeat-x; background-position: 0px 160px;">
<form style="margin: 0px; margin-top:30px; padding:0px;" id="form"
action="ricerca_comune.php" target="" method="POST" onKeyPress="">
    <input type="hidden" id="valore_provincia" name="valore_provincia" value
="<?php echo $cod_provincia ?>" style="display:none; width:0px" />
    <input type="hidden" id="valore_comune" name="valore_comune" value
="<?php echo $cod_comune ?>" style="display:none; width:0px" />
    <table>
    <tr>
    <td width="100px">
        <label class="reportStile1">Provincia:</label>
    </td>
    <td width="200px">

```

```

        <input type="text" id="provincia" name="provincia" style="margin-
left: 0px; width: 170px; display: inline" value="<?php echo $provincia ?>"
/>
        
        <div id="hint_provincia" class="autocomplete provincia"></div>
        <script>
            var params = "";

            //debugger;
            var element_provincia =
document.getElementById("valore_provincia");

            params = "SESSION=" + encodeURIComponent(session) +
"&CHECKSESSION=1";
            new Ajax.Autocompleter("provincia", "hint_provincia",
"ricerca_comune_provincia_action.php", {afterUpdateElement:getProvincia,param
eters:params});
            function getProvincia(text, li)
            {
                element_provincia.value = li.id;
                document.getElementById("provincia").value = li.innerHTML;
                document.getElementById("form").submit();
            }
        </script>
    </td>
</tr>

<tr>
    <td width="100px">
        <label class="reportStile1">Comune:</label>
    </td>
    <td width="200px">
        <input type="text" id="comune" name="comune" style="margin-left:
0px; width: 170px; display: inline" value="<?php echo $comune ?>" />
        \'' )" />
    <div id="hint_comune" class="autocomplete comune"></div>
    <script>
        var params = "";

        //debugger;
        var element_provincia =
document.getElementById("valore_provincia");
        var element_comune = document.getElementById("valore_comune");

        params = "SESSION=" + encodeURIComponent(session) +
"&CHECKSESSION=1&PROVINCIA=" + encodeURIComponent(element_provincia.value);
        new Ajax.Autocompleter("comune", "hint_comune",
"ricerca_comune_comune_action.php",{afterUpdateElement:getComune,parameters:
params});
        function getComune(text, li)
        {
            element_comune.value = li.id;
            document.getElementById("comune").value = li.innerHTML;
            document.getElementById("form").submit();
        }
    </script>
</td>
</tr>

<tr>
    <td height="100" colspan="2" valign="top">
        <div id="listBlock" style="display:none">
            <select style="display: inline; float: left; margin-right:
5px; margin-left: 5px; width: 85%; height: 95px" name="listBox" id="listBox"
size="5" multiple="multiple" onchange="changeValue(this.value)">
                </select>
            
        </div>
    </td>
</tr>

<tr align="right">
    <td colspan="2" style="padding-bottom: 10px;">

```

```

        <input type="button" value="Cerca" onclick="cerca();
parent.nascondi()" class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
        <input type="button" value="Chiudi" onclick="parent.nascondi()"
class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />

    </td>
</tr>
</table>
</form>
</body>
</html>

```

### **5.6.2.3 ricerca\_comune\_provincia\_getList.php**

```

<?php
session_start();
require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'DbHanFDO.php';
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

$json = new Services_JSON();
$responseType = '';
$response = '';

$dbHan = new DbHanFDO($args,$args['SESSION'],'Map');
$responseType = 'text/plain';

$response = $json->encode($dbHan->Select('ORDER07,NAME', 'PROVINCE', null,
1, 'NAME,ASC'));
unset($dbHan);

header('Content-Type: ' . $responseType);
echo $response;
?>

```

### **5.6.2.4 ricerca\_comune\_provincia\_action.php**

```

<?php
session_start();
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'DbHanFDO.php';

```

```

require_once '../' . $frameworkPath . 'DbHanDBMS.php';
session_id($args['SESSION']);

$provincia = $args['provincia'];

// Estraggo dal catalogo delle province tutte le province che iniziano per
il valore dato.
$dbHan = new DbHanFDO($args, $args['SESSION'], 'Map');
$where = "NAME LIKE '" . $provincia . "%'";
$response = $dbHan->Select('ORDER07,NAME', 'PROVINCE', $where, 1,
'NAME,ASC');
unset($dbHan);
?>
<ul>
<?php
    if(count($response) != 0)
    {
        $max = count($response);
        if($max > 10)
            $max = 10;
        for($i=0; $i<$max; $i++)
        {
            $str = "<li id='" . $response[$i][0] . "'>" . $response[$i][1] .
"</li>";
            echo $str;
        }
    }
?>
</ul>

```

#### **5.6.2.5 ricerca\_comune\_comune\_getList.php**

```

<?php
session_start();
require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'DbHanFDO.php';
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

$json = new Services_JSON();
$responseType = '';
$response = '';

$dbHan = new DbHanFDO($args,$args['SESSION'],'Map');
$responseType = 'text/plain';

$response = $json->encode($dbHan->Select('ORDER08,NAME', 'COMUNI',
$args['WHERECLAUSE'], 1, 'NAME,ASC'));
unset($dbHan);

```

```
header('Content-Type: ' . $responseType);
echo $response;
?>
```

### **5.6.2.6 ricerca\_comune\_comune\_action.php**

```
<?php
session_start();
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'DbHanFDO.php';
require_once '../' . $frameworkPath . 'DbHanDBMS.php';
session_id($args['SESSION']);

$comune = $args['comune'];
$ISTAT_provincia = $args['PROVINCIA'];

// Estraggo dal catalogo dei comuni tutti i comuni che iniziano per il
valore dato.
$dbHan = new DbHanFDO($args, $args['SESSION'], 'Map');
$where = "(NAME LIKE '" . $comune .
"%')AND(ORDER07='" . $ISTAT_provincia . "'";
$response = $dbHan->Select('ORDER08,NAME', 'COMUNI', $where, 1, 'NAME,ASC');
unset($dbHan);
?>
<ul>
<?php
    if(count($response) != 0)
    {
        $max = count($response);
        if($max > 10)
            $max = 10;
        for($i=0; $i<$max; $i++)
        {
            $str = "<li id='" . $response[$i][0] . "'>" . $response[$i][1] .
"</li>";
            echo $str;
        }
    }
?>
</ul>
```

### **5.6.2.7 ricerca\_comune\_indirizzo.php**

```
<?php
session_start();
require_once '../include/siteconstants.php';
```

```

require_once '../' . $frameworkPath . 'AbFmw.php';
require_once '../' . $frameworkPath . 'DbHanFDO.php';
require_once '../' . $frameworkPath . 'DbHanDBMS.php';

require_once '../' . $frameworkPath . 'js/commonjs.php';
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

$provincia = "";
$cod_provincia = $args['valore_provincia'];

$dbHan = new DbHanFDO($args, $_SESSION['MgSessionId'], 'Map');
$where = "ORDER07='" . $cod_provincia . "'";
$response = $dbHan->Select('NAME', 'PROVINCE', $where, 1);
unset($dbHan);

$provincia = $response[0][0];

$comune = "";
$cod_comune = $args['valore_comune'];

$dbHan = new DbHanFDO($args, $_SESSION['MgSessionId'], 'Map');
$where = "ORDER08='" . $cod_comune . "'";
$response = $dbHan->Select('NAME', 'COMUNI', $where, 1);
unset($dbHan);

$comune = $response[0][0];

$indirizzo = "";
$cod_indirizzo = $args['valore_indirizzo'];
$indirizzo = $cod_indirizzo;
?>
<html>
  <head>
    <title><?php $title?></title>
    <link href="../../../Stili.css" rel="stylesheet" type="text/css">

    <style>
      .autocomplete
      {
        font-family: Tahoma, Arial, Helvetica, sans-serif;
        font-size: 8pt;
        position: absolute;
        background-color: white;
        border: 1px solid #888;
        margin: 0;
        padding: 0;
      }
  </head>
</html>

```

```

.autocomplete ul
{
  list-style-type:none;
  margin:0;
  padding:0;
}

.autocomplete ul li.selected
{
  background-color: #FFB;
}
.autocomplete ul li
{
  list-style-type:none;
  display:block;
  margin:0;
  padding:2px;
  cursor:pointer;
}
</style>

<script type="text/javascript" src="..<?php echo $frameworkPath
?>js/prototype.js"></script>
<script type="text/javascript" src="..<?php echo $frameworkPath
?>js/effects.js"></script>
<script type="text/javascript" src="..<?php echo $frameworkPath
?>js/controls.js"></script>
<script language="javascript"
src="../../../../viewerfiles/browserdetect.js"></script>
<script language="javascript" src="..<?php echo $frameworkPath
?>js/json.js"></script>
<script language="javascript"
src="../../../../moduli/modulo_base/js/highlight_ricerca.js"></script>

<script language = "JavaScript" type = "text/JavaScript">
var session = '<?php echo $_SESSION['MgSessionId'] ?>';
var mapName = 'Map';
var caller = "";
var cod_provincia = "<?php echo $cod_provincia ?>";
var cod_comune = "<?php echo $cod_comune ?>";
var cod_indirizzo = "<?php echo $cod_indirizzo?>";

var mapFrame =
parent.document.getElementById("indexFrame").contentWindow.document.getEleme
ntById("mgFrame").contentWindow.document.getElementById("mapFrame");

```



```

var viewerFrame =
parent.document.getElementById("indexFrame").contentWindow.document.getEleme
ntById("mgFrame");

function getList(element,whereClause)
{
var list = document.getElementById("listBox");
//debugger;
var reqParams = "SESSION=" + encodeURIComponent(session);

var nomeFile="";
switch(element){
case 'valore_provincia':
nomeFile = "ricerca_comune_provincia_getList.php";
break;
case 'valore_comune':
reqParams += "&WHERECLAUSE=" + encodeURIComponent(whereClause);
nomeFile = "ricerca_comune_comune_getList.php";
break;
case 'valore_indirizzo':
reqParams += "&WHERECLAUSE=" + encodeURIComponent(whereClause);
nomeFile = "ricerca_comune_indirizzo_indirizzo_getList.php";
break;
}

if (msie)
reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
else
reqHandler = new XMLHttpRequest();

reqHandler.open("POST", nomeFile, false);
reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

reqHandler.send(reqParams);

var values = reqHandler.responseText.parseJSON();

for(var i=0; i<values.length; i++)
{
list_el = null;
list_el = document.createElement("option");

list_el.value = values[i][0];
list_el.innerHTML = values[i][1];

list.appendChild(list_el);
}
}

```

```

    }

}

function cerca()
{
    //debugger;
    var reqParams = "COMMAND=SELECT";
    reqParams += "&SESSION=" + encodeURIComponent(session);
    reqParams += "&NOMECAPISTR=" + encodeURIComponent("ID");
    reqParams += "&NOMETABELLESTR=" + encodeURIComponent("ita_gc");
    reqParams += "&WHERECLAUSE=";

    var whereClause = "(L_ORDER08='" + encodeURIComponent(cod_comune) +
    "' ) OR (R_ORDER08='" + encodeURIComponent(cod_comune) + "' )AND";
    whereClause += "(FULLNAME='" + cod_indirizzo + "' )";

    reqParams += encodeURIComponent(whereClause);

    if (msie)
        reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
    else
        reqHandler = new XMLHttpRequest();

    reqHandler.open("POST", "dbhandbms_controller.php", false);
    reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
    urlencoded");

    reqHandler.send(reqParams);

    //debugger;
    var values = reqHandler.responseText.parseJSON();

    activateLayer("RETE");

    var ids = values.join(','); //per eseguire la query
    var idsStr = values[0]+""; //per eseguire l'highlight
    for(var i=1; i < values.length; i++){
        idsStr += " OR ID=" + values[i];
    }

    reqParams = "COMMAND=GETENVELOPE";
    reqParams += "&SESSION=" + encodeURIComponent(session);
    reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
    reqParams += "&LAYERNAME=RETE";
    reqParams += "&IDFIELD=ID";
    reqParams += "&IDS=" + encodeURIComponent(ids);
}

```

```

    if (msie)
        reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
    else
        reqHandler = new XMLHttpRequest();

    reqHandler.open("POST",
"..../moduli/modulo_base/fdo/abfmw_controller.php", false);
    reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

    reqHandler.send(reqParams);

    envelope = reqHandler.responseText.parseJSON();

    if (envelope != "ENVELOPE IS NULL")
        ZoomSimple_new(envelope, mapFrame);
    else
        alert ('Indirizzo non trovato!');

    HighlightAllOff();
    //debugger;
    Highlight(idsStr, "ID", "RETE");

    // Consente l'apertura del report.
    //window.open("../report_controller.php?KEYS=' '&TABLES=RETE&SESSION="
+ session + "&clausolaWhere=" +
encodeURIComponent(whereClause), "report1Frame");
}

function activateLayer(name)
{
    var reqParams = "COMMAND=ACTIVATELAYER";
    reqParams += "&SESSION=" + encodeURIComponent(session);
    reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
    reqParams += "&LAYERNAME=" + encodeURIComponent(name);

    if (msie)
        reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
    else
        reqHandler = new XMLHttpRequest();

    reqHandler.open("POST",
"..../moduli/modulo_base/fdo/abfmw_controller.php", false);
    reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

```

```

    reqHandler.send(reqParams);

    viewerFrame.contentWindow.Refresh();
}

function changeValue(codice)
{
    document.getElementById(caller).value = codice;
    document.getElementById("form").submit();
}
</script>
</head>
<body style="padding:0; margin: 0; background: white; background-image:
url('../..../moduli/modulo_base/image/dimming/pf_report_bottom.jpg');
background-repeat: repeat-x; background-position: 0px 190px;">
    <form style="margin: 0px; margin-top:30px; padding:0px;" id="form"
action="ricerca_comune_indirizzo.php" target="" method="POST" onKeyPress="">
        <input type="hidden" id="valore_provincia" name="valore_provincia" value
="<?php echo $cod_provincia ?>" style="display:none; width:0px" />
        <input type="hidden" id="valore_comune" name="valore_comune" value
="<?php echo $cod_comune ?>" style="display:none; width:0px" />
        <input type="hidden" id="valore_indirizzo" name="valore_indirizzo" value
="<?php echo $cod_indirizzo ?>" style="display:none; width:0px" />
        <table>
            <tr>
                <td width="100px">
                    <label class="reportStile1">Provincia:</label>
                </td>
                <td width="200px">
                    <input type="text" id="provincia" name="provincia" style="margin-
left: 0px; width: 170px; display: inline" value="<?php echo $provincia ?>"
/>
                    
                    <div id="hint_provincia" class="autocomplete provincia"></div>
                </td>
            </tr>
        </table>
        <script>
            var params = "";

            //debugger;

```

```

        var element_provincia =
document.getElementById("valore_provincia");

        params = "SESSION=" + encodeURIComponent(session) +
"&CHECKSESSION=1";
        new Ajax.Autocompleter("provincia", "hint_provincia",
"ricerca_comune_provincia_action.php",{afterUpdateElement:getProvincia,param
eters:params});
        function getProvincia(text, li)
        {
            element_provincia.value = li.id;
            document.getElementById("provincia").value = li.innerHTML;
            document.getElementById("form").submit();
        }
    </script>
</td>
</tr>

<tr>
    <td width="100px">
        <label class="reportStile1">Comune:</label>
    </td>
    <td width="200px">
        <input type="text" id="comune" name="comune" style="margin-left:
0px; width: 170px; display: inline" value="<?php echo $comune ?>" />
        \'')" />
        <div id="hint_comune" class="autocomplete_comune"></div>
    <script>
        var params = "";

        //debugger;
        var element_provincia =
document.getElementById("valore_provincia");
        var element_comune = document.getElementById("valore_comune");

        params = "SESSION=" + encodeURIComponent(session);
        params += "&CHECKSESSION=1&PROVINCIA=" +
encodeURIComponent(element_provincia.value);

```

```

        new Ajax.Autocompleter("comune", "hint_comune",
"ricerca_comune_comune_action.php",{afterUpdateElement:getComune,parameters:
params});
        function getComune(text, li)
        {
            element_comune.value = li.id;
            document.getElementById("comune").value = li.innerHTML;
            document.getElementById("form").submit();
        }
    </script>
</td>
</tr>

<tr>
    <td width="100px">
        <label class="reportStile1">Indirizzo:</label>
    </td>
    <td width="200px">
        <input type="text" id="indirizzo" name="indirizzo" style="margin-
left: 0px; width: 170px; display: inline" value="<?php echo $indirizzo ?>"
/>
        \') OR (R_ORDER08='\<?php echo $cod_comune?>\'))'" />
        <div id="hint_indirizzo" class="autocomplete indirizzo"></div>
        <script>
            var params = "";

            //debugger;
            var element_comune = document.getElementById("valore_comune");
            var element_indirizzo =
document.getElementById("valore_indirizzo");

            params = "SESSION=" + encodeURIComponent(session);
            params += "&CHECKSESSION=1&COMUNE=" +
encodeURIComponent(element_comune.value);
            new Ajax.Autocompleter("indirizzo", "hint_indirizzo",
"ricerca_comune_indirizzo_indirizzo_action.php",{afterUpdateElement:getIndir
izzo,parameters:params});
            function getIndirizzo(text, li)
            {

```

```

        element_indirizzo.value = li.id;
        document.getElementById("indirizzo").value = li.innerHTML;
        document.getElementById("form").submit();
    }
</script>
</td>
</tr>

<tr>
    <td height="100" colspan="2" valign="top">
        <div id="listBlock" style="display:none">
            <select style="display: inline; float: left; margin-right:
5px; margin-left: 5px; width: 85%; height: 95px" name="listBox" id="listBox"
size="5" multiple="multiple" onchange="changeValue(this.value)">
                </select>
                
            </div>
        </td>
</tr>

<tr align="right">
    <td colspan="2" style="padding-bottom: 10px;">
        <input type="button" value="Cerca" onclick="cerca();
parent.nascondi()" class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
        <input type="button" value="Chiudi" onclick="parent.nascondi()"
class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
    </td>
</tr>
</table>
</form>
</body>
</html>

```

### 5.6.2.8 *ricerca\_comune\_indirizzo\_indirizzo\_getList.php*

```
<?php
```

```

session_start();
require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'DbHanDBMS.php';
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

$json = new Services_JSON();
$responseType = '';
$response = array();

$dbConn = connectTo('mgos', $dbConnSet);

$dbHan = new
DbHanDBMS($dbConn, $dbConnSet['mgos']['DBMS'], $args, $args['SESSION']);
$responseType = 'text/plain';
$response = $json->encode($dbHan->Select('FULLNAME AS INDEX, FULLNAME',
'ita_gc', $args['WHERECLAUSE'], 1, 'FULLNAME, ASC'));
unset($dbHan);

$dbConn -> Close();

header('Content-Type: ' . $responseType);
echo $response;
?>

```

### **5.6.2.9 ricerca\_comune\_indirizzo\_indirizzo\_action.php**

```

<?php
session_start();
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'DbHanFDO.php';
require_once '../' . $frameworkPath . 'DbHanDBMS.php';
session_id($args['SESSION']);

$indirizzo = $args['indirizzo'];
$ISTAT_comune = $args['COMUNE'];

// Estraggo dal catalogo teleatlas gli indirizzi del paese dato che iniziano
per il valore dato.
$dbConn = connectTo('mgos', $dbConnSet);

$dbHan = new DbHanDBMS($dbConn, $dbConnSet['mgos']['DBMS'], $args,
$args['SESSION']);
$where = "((L_ORDER08='". $ISTAT_comune. "') OR
(R_ORDER08='". $ISTAT_comune. "))AND(FULLNAME LIKE '" . $indirizzo . "%)";
$response = $dbHan->Select('FULLNAME', 'ita_gc', $where, 1, 'FULLNAME, ASC');
unset($dbHan);

```



```

$dbConn -> Close();
?>
<ul>
<?php
    if(count($response) != 0)
    {
        $max = count($response);
        if($max > 10)
            $max = 10;
        for($i=0; $i<$max; $i++)
        {
            $str = "<li id='" . $response[$i][0] . "'>" . $response[$i][0] .
"</li>";
            echo $str;
        }
    }
?>
</ul>

```

#### **5.6.2.10 ricerca\_comune\_indirizzo\_civico.php**

```

<?php
session_start();
require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'AbFmw.php';
require_once '../' . $frameworkPath . 'DbHanFDO.php';
require_once '../' . $frameworkPath . 'DbHanDBMS.php';

require_once '../' . $frameworkPath . 'js/commonjs.php';
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

$provincia = "";
$cod_provincia = $args['valore_provincia'];

$dbHan = new DbHanFDO($args, $_SESSION['MgSessionId'], 'Map');
$where = "ORDER07='" . $cod_provincia . "'";
$response = $dbHan->Select('NAME', 'PROVINCE', $where, 1);
unset($dbHan);

$provincia = $response[0][0];

$comune = "";
$cod_comune = $args['valore_comune'];

$dbHan = new DbHanFDO($args, $_SESSION['MgSessionId'], 'Map');
$where = "ORDER08='" . $cod_comune . "'";
$response = $dbHan->Select('NAME', 'COMUNI', $where, 1);

```

```

unset($dbHan);

$comune = $response[0][0];

$cod_indirizzo = $args['valore_indirizzo'];

$civico = $args['civico'];
$cod_civico = $args['valore_civico'];
?>
<html>
  <head>
    <title><?php $title?></title>
    <link href="../../Stili.css" rel="stylesheet" type="text/css">

    <style>
      .autocomplete
      {
        font-family: Tahoma, Arial, Helvetica, sans-serif;
        font-size: 8pt;
        position:absolute;
        background-color:white;
        border:1px solid #888;
        margin:0;
        padding:0;
      }

      .autocomplete ul
      {
        list-style-type:none;
        margin:0;
        padding:0;
      }

      .autocomplete ul li.selected
      {
        background-color: #FFB;
      }

      .autocomplete ul li
      {
        list-style-type:none;
        display:block;
        margin:0;
        padding:2px;
        cursor:pointer;
      }
    </style>

```

```

    <script type="text/javascript" src="../../?php echo $frameworkPath
?>js/prototype.js"></script>
    <script type="text/javascript" src="../../?php echo $frameworkPath
?>js/effects.js"></script>
    <script type="text/javascript" src="../../?php echo $frameworkPath
?>js/controls.js"></script>
    <script language="javascript"
src="../../../../../../viewerfiles/browserdetect.js"></script>
    <script language="javascript" src="../../?php echo $frameworkPath
?>js/json.js"></script>
    <script language="javascript"
src="../../moduli/modulo_base/js/highlight_ricerca.js"></script>

<script language = "JavaScript" type = "text/JavaScript">
var session = '<?php echo $_SESSION['MgSessionId'] ?>';
var mapName = 'Map';
var caller = "";
var cod_provincia = "<?php echo $cod_provincia ?>";
var cod_comune = "<?php echo $cod_comune ?>";
var cod_indirizzo = "<?php echo $cod_indirizzo?>";

var mapFrame =
parent.document.getElementById("indexFrame").contentWindow.document.getEleme
ntById("mgFrame").contentWindow.document.getElementById("mapFrame");
var viewerFrame =
parent.document.getElementById("indexFrame").contentWindow.document.getEleme
ntById("mgFrame");

function getList(element,whereClause)
{
    var list = document.getElementById("listBox");
    //debugger;
    var reqParams = "SESSION=" + encodeURIComponent(session);

    var nomeFile="";
    switch(element){
    case 'valore_provincia':
        nomeFile = "ricerca_comune_provincia_getList.php";
        break;
    case 'valore_comune':
        reqParams += "&WHERECLAUSE=" + encodeURIComponent(whereClause);
        nomeFile = "ricerca_comune_comune_getList.php";
        break;
    case 'valore_indirizzo':
        reqParams += "&WHERECLAUSE=" + encodeURIComponent(whereClause);
        nomeFile = "ricerca_comune_indirizzo_indirizzo_getList.php";
        break;

```

```

    }

    if (msie)
        reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
    else
        reqHandler = new XMLHttpRequest();

    reqHandler.open("POST", nomeFile, false);
    reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

    reqHandler.send(reqParams);

    var values = reqHandler.responseText.parseJSON();

    for(var i=0; i<values.length; i++)
    {
        list_el = null;
        list_el = document.createElement("option");

        list_el.value = values[i][0];
        list_el.innerHTML = values[i][1];

        list.appendChild(list_el);
    }
}

function cerca()
{
    debugger;
    var numCivico = "";
    var civico = document.getElementById("civico").value;
    for(var i=0; i<civico.length; i++){
        if((civico.charAt(i)>'0')&&(civico.charAt(i)<='9')){
            numCivico += civico.charAt(i);
        }
    }

    var reqParams = "COMMAND=SELECT";
    reqParams += "&SESSION=" + encodeURIComponent(session);
    reqParams += "&NOMECAMPISTR=" +
encodeURIComponent("ID,L_F_ADD,L_T_ADD,R_F_ADD,R_T_ADD");
    reqParams += "&NOMETABELLESTR=" + encodeURIComponent("ita_gc");
    reqParams += "&WHERECLAUSE=";

```

```

    var whereClause = "(L_ORDER08='" + encodeURIComponent(cod_comune) +
    "') OR (R_ORDER08='" + encodeURIComponent(cod_comune) + "')AND";
    whereClause += "(FULLNAME='" + cod_indirizzo + "') AND NOT(L_F_ADD =-1
    AND R_F_ADD = -1)";

    reqParams += encodeURIComponent(whereClause);

    if (msie)
        reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
    else
        reqHandler = new XMLHttpRequest();

    reqHandler.open("POST", "dbhandbms_controller.php", false);
    reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
    urlencoded");

    reqHandler.send(reqParams);

    //debugger;
    var values = reqHandler.responseText.parseJSON();

    if(values != false){
        //sol: side of line ovvero sinistra 1 destra 3
        var sol = null;
        for(var i = 0; i < values.length && sol==null; i++){
            if(values[i][1] != -1 && values[i][1]%2 == numCivico%2){
                sol = 1;
            }else if(values[i][3] != -1 && values[i][3]%2 == numCivico%2){
                sol = 3;
            }
        }
        //prima devo ordinare i civici
        //sol indica il campo secondo cui ordinare
        values = bubbleSort(values,sol,(sol==1) ? 3 : 1);

        var featId;
        var trovato = false;
        if(sol != null){
            for(var i = 0; i < values.length && !trovato; i++){
                var minCivico;
                var maxCivico;
                if (parseInt(values[i][sol]) <= parseInt(values[i][sol + 1])){
                    minCivico = parseInt(values[i][sol]);
                    maxCivico = parseInt(values[i][sol + 1]);
                }
                else{
                    minCivico = parseInt(values[i][sol + 1]);
                }
            }
        }
    }

```

```

        maxCivico = parseInt(values[i][sol]);
    }

    if(numCivico <= maxCivico || i == values.length){
        //se il civico cercato appartiene al tronchetto corrente
        //allora restituisco l'id
        featID = values[i][0];
        trovato = true;
    }
}
}

activateLayer("RETE");

//zoom sul tronchetto dove si trova il civico
reqParams = "COMMAND=GETENVELOPE";
reqParams += "&SESSION=" + encodeURIComponent(session);
reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
reqParams += "&LAYERNAME=RETE";
reqParams += "&IDFIELD=ID";
reqParams += "&IDS=" + encodeURIComponent(featID);

if (msie)
    reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
else
    reqHandler = new XMLHttpRequest();

reqHandler.open("POST",
"..../moduli/modulo_base/fdo/abfmw_controller.php", false);
reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

reqHandler.send(reqParams);

envelope = reqHandler.responseText.parseJSON();

if (envelope != "ENVELOPE IS NULL")
    ZoomSimple_new(envelope, mapFrame);
else
    alert ('Civico non trovato!');

HighlightAllOff();
//niente highlight in quanto faccio lo zoom sul tronchetto e non sul
civico
//debugger;

```

```

        //Highlight(ids, "ID", "RETE");

        // Consente l'apertura del report.

        //window.open("../report_controller.php?KEYS='&TABLES=RETE&SESSION=" +
session + "&clausolaWhere=" +
encodeURIComponent(whereClause),"report1Frame");
    }
    else
        alert ('Indirizzo non trovato!');
}

function activateLayer(name)
{
    var reqParams = "COMMAND=ACTIVATELAYER";
    reqParams += "&SESSION=" + encodeURIComponent(session);
    reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
    reqParams += "&LAYERNAME=" + encodeURIComponent(name);

    if (msie)
        reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
    else
        reqHandler = new XMLHttpRequest();

    reqHandler.open("POST",
"../../moduli/modulo_base/fdo/abfmw_controller.php", false);
    reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

    reqHandler.send(reqParams);

    viewerFrame.contentWindow.Refresh();
}

function changeValue(codice)
{
    document.getElementById(caller).value = codice;
    document.getElementById("form").submit();
}

//array: l'array bidimensionale da ordinare
//index: l'indice del primo campo secondo cui ordinare
//index2: l'indice del secondo campo secondo cui ordinare
function bubbleSort(array,index,index2){
    var ordinato;
    do{
        ordinato = true;

```

```

        for(var i=1; i<array.length; i++){
            if(parseInt(array[i][index]) < parseInt(array[i-1][index]) ||
(parseInt(array[i][index]) == parseInt(array[i-
1][index])&&parseInt(array[i][index2]) < parseInt(array[i-1][index2]))) {
                var temp = array[i];
                array[i]=array[i-1];
                array[i-1]=temp;
                ordinato = false;
            }
        }
    }while(!ordinato);
    return array;
}
</script>
</head>
<body style="padding:0; margin: 0; background: white; background-image:
url('../..../moduli/modulo_base/image/dimming/pf_report_bottom.jpg');
background-repeat: repeat-x; background-position: 0px 220px;">
    <form style="margin: 0px; margin-top:30px; padding:0px;" id="form"
action="ricerca_comune_indirizzo_civico.php" target="" method="POST"
onKeyPress="">
        <input type="hidden" id="valore_provincia" name="valore_provincia" value
="<?php echo $cod_provincia ?>" style="display:none; width:0px" />
        <input type="hidden" id="valore_comune" name="valore_comune" value
="<?php echo $cod_comune ?>" style="display:none; width:0px" />
        <input type="hidden" id="valore_indirizzo" name="valore_indirizzo" value
="<?php echo $cod_indirizzo ?>" style="display:none; width:0px" />
        <input type="hidden" id="valore_civico" name="valore_civico" value
="<?php echo $cod_civico ?>" style="display:none; width:0px" />
        <table>
            <tr>
                <td width="100px">
                    <label class="reportStile1">Provincia:</label>
                </td>
                <td width="200px">
                    <input type="text" id="provincia" name="provincia" style="margin-
left: 0px; width: 170px; display: inline" value="<?php echo $provincia ?>"
/>
                    
                    <div id="hint_provincia" class="autocomplete provincia"></div>

```



```

<script>
    var params = "";

    //debugger;
    var element_provincia =
document.getElementById("valore_provincia");

        params = "SESSION=" + encodeURIComponent(session) +
"&CHECKSESSION=1";
        new Ajax.Autocompleter("provincia", "hint_provincia",
"ricerca_comune_provincia_action.php",{afterUpdateElement:getProvincia,param
eters:params});
        function getProvincia(text, li)
        {
            element_provincia.value = li.id;
            document.getElementById("provincia").value = li.innerHTML;
            document.getElementById("form").submit();
        }
    </script>
</td>
</tr>

<tr>
    <td width="100px">
        <label class="reportStile1">Comune:</label>
    </td>
    <td width="200px">
        <input type="text" id="comune" name="comune" style="margin-left:
0px; width: 170px; display: inline" value="<?php echo $comune ?>" />
        \'')" />
        <div id="hint_comune" class="autocomplete_comune"></div>
    <script>
        var params = "";

        //debugger;
        var element_provincia =
document.getElementById("valore_provincia");
        var element_comune = document.getElementById("valore_comune");

```

```

        params = "SESSION=" + encodeURIComponent(session);
        params += "&CHECKSESSION=1&PROVINCIA=" +
encodeURIComponent(element_provincia.value);
        new Ajax.Autocompleter("comune", "hint_comune",
"ricerca_comune_comune_action.php",{afterUpdateElement:getComune,parameters:
params});
        function getComune(text, li)
        {
            element_comune.value = li.id;
            document.getElementById("comune").value = li.innerHTML;
            document.getElementById("form").submit();
        }
    </script>
</td>
</tr>

<tr>
    <td width="100px">
        <label class="reportStile1">Indirizzo:</label>
    </td>
    <td width="200px">
        <input type="text" id="indirizzo" name="indirizzo" style="margin-
left: 0px; width: 170px; display: inline" value="<?php echo $cod_indirizzo
?>" />
        \') OR (R_ORDER08='\<?php echo $cod_comune?>\'))'" />
        <div id="hint_indirizzo" class="autocomplete indirizzo"></div>
        <script>
            var params = "";

            //debugger;
            var element_comune = document.getElementById("valore_comune");
            var element_indirizzo =
document.getElementById("valore_indirizzo");

            params = "SESSION=" + encodeURIComponent(session);
            params += "&CHECKSESSION=1&COMUNE=" +
encodeURIComponent(element_comune.value);

```

```

        new Ajax.Autocompleter("indirizzo", "hint_indirizzo",
"ricerca_comune_indirizzo_indirizzo_action.php",{afterUpdateElement:getIndir
izzo,parameters:params});
        function getIndirizzo(text, li)
        {
            element_indirizzo.value = li.id;
            document.getElementById("indirizzo").value = li.innerHTML;
            document.getElementById("form").submit();
        }
    </script>
</td>
</tr>

<tr>
    <td width="100px">
        <label class="reportStile1">Civico:</label>
    </td>
    <td width="200px">
        <input type="text" id="civico" name="civico" style="margin-left:
0px; width: 170px; display: inline" value="<?php echo $civico ?>"/>
        <!-- Niente getList e autocompleter in quanto non è possibile
realizzarli in modo efficace -->
    </td>
</tr>

<tr>
    <td height="100" colspan="2" valign="top">
        <div id="listBlock" style="display:none">
            <select style="display: inline; float: left; margin-right:
5px; margin-left: 5px; width: 85%; height: 95px" name="listBox" id="listBox"
size="5" multiple="multiple" onchange="changeValue(this.value)">
                </select>
            
        </div>
    </td>
</tr>

<tr align="right">
    <td colspan="2" style="padding-bottom: 10px;">
        <input type="button" value="Cerca" onclick="cerca();
parent.nascondi()" class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';

```

```

style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
        <input type="button" value="Chiudi" onclick="parent.nascondi()"
class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />

        </td>
    </tr>
</table>
</form>
</body>
</html>

```

### 5.6.2.11 ricerca\_cliente.php

```

<?php
session_start();
require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'AbFmw.php';
require_once '../' . $frameworkPath . 'DbHanDBMS.php';
require_once '../' . $frameworkPath . 'DbHanFDO.php';

require_once '../' . $frameworkPath . 'js/commonjs.php';
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

$cliente="";
$cod_cliente = $args['valore_cliente'];

$dbConnCRM = connectTo('crm',$dbConnSet);

$dbHan = new DbHanDBMS($dbConnCRM, $dbConnSet['crm']['DBMS'], $args,
$_SESSION['MgSessionId']);
$where = "ID='".$cod_cliente.'"";
$response = $dbHan->Select('NAME','ACCOUNTS',$where,1);
unset($dbHan);

$dbConnCRM -> Close();

$cliente = $response[0][0];
?>
<html>
    <head>
        <title><?php $title?></title>
        <link href="../Stili.css" rel="stylesheet" type="text/css">

        <style>

```

```

.autocomplete
{
    font-family: Tahoma, Arial, Helvetica, sans-serif;
    font-size: 8pt;
    position: absolute;
    background-color: white;
    border: 1px solid #888;
    margin: 0;
    padding: 0;
}

.autocomplete ul
{
    list-style-type: none;
    margin: 0;
    padding: 0;
}

.autocomplete ul li.selected
{
    background-color: #FFB;
}

.autocomplete ul li
{
    list-style-type: none;
    display: block;
    margin: 0;
    padding: 2px;
    cursor: pointer;
}
</style>

<script type="text/javascript" src="../../?php echo $frameworkPath
?>js/prototype.js"></script>
<script type="text/javascript" src="../../?php echo $frameworkPath
?>js/effects.js"></script>
<script type="text/javascript" src="../../?php echo $frameworkPath
?>js/controls.js"></script>
<script language="javascript"
src="../../../../viewerfiles/browserdetect.js"></script>
<script language="javascript" src="../../?php echo $frameworkPath
?>js/json.js"></script>
<script language="javascript"
src="../../moduli/modulo_base/js/highlight_ricerca.js"></script>

<script language="JavaScript" type="text/JavaScript">
var session = '<?php echo $_SESSION['MgSessionId'] ?>';

```

```

var mapName = 'Map';
var caller = "";

var cod_cliente = '<?php echo $cod_cliente ?>';

var mapFrame =
parent.document.getElementById("indexFrame").contentWindow.document.getEleme
ntById("mgFrame").contentWindow.document.getElementById("mapFrame");
var viewerFrame =
parent.document.getElementById("indexFrame").contentWindow.document.getEleme
ntById("mgFrame");

function getList()
{
var list = document.getElementById("listBox");

var reqParams = "SESSION=" + encodeURIComponent(session);

if (msie)
reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
else
reqHandler = new XMLHttpRequest();

reqHandler.open("POST", "ricerca_cliente_getList.php", false);
reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

reqHandler.send(reqParams);

//debugger;
var values = reqHandler.responseText.parseJSON();

for(var i=0; i<values.length; i++)
{
list_el = null;
list_el = document.createElement("option");

list_el.value = values[i][0];
list_el.innerHTML = values[i][1];

list.appendChild(list_el);
}
}

function cerca()
{

```

```

//debugger;
var reqParams = "COMMAND=SELECTFEAUTURES";
reqParams += "&SESSION=" + encodeURIComponent(session);
reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
reqParams += "&LAYERNAME=" + encodeURIComponent("AZIENDE");
reqParams += "&LAYERPROPERTIES=" + encodeURIComponent("FeatId");
reqParams += "&WHERECLAUSE=";

var whereClause = "(COD_AZIENDA='" + encodeURIComponent(cod_cliente) +
''');

reqParams += encodeURIComponent(whereClause);

if (msie)
    reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
else
    reqHandler = new XMLHttpRequest();

reqHandler.open("POST",
"../../moduli/modulo_base/fdo/abfmw_controller.php", false);
reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

reqHandler.send(reqParams);

//debugger;
var values = reqHandler.responseText.parseJSON();
var ids = "";

ids = encodeURIComponent(values[0].FeatId);

activateLayer("AZIENDE");

reqParams = "COMMAND=GETENVELOPE";
reqParams += "&SESSION=" + encodeURIComponent(session);
reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
reqParams += "&LAYERNAME=AZIENDE";
reqParams += "&IDFIELD=FeatId";
reqParams += "&IDS=" + encodeURIComponent(ids);

if (msie)
    reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
else
    reqHandler = new XMLHttpRequest();

reqHandler.open("POST",
"../../moduli/modulo_base/fdo/abfmw_controller.php", false);

```

```

    reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

    reqHandler.send(reqParams);

    envelope = reqHandler.responseText.parseJSON();

    if (envelope != "ENVELOPE IS NULL")
        ZoomSimple_new(envelope, mapFrame);
    else
        alert ('Cliente non Trovato!');

    HighlightAllOff();
    Highlight(values[0].FeatId, "FeatId", "AZIENDE");

    // Consente l'apertura del report.
    window.open("../report_controller.php?KEYS=''&TABLES=AZIENDE&SESSION="
+ session + "&clausolaWhere=" +
encodeURIComponent(whereClause),"report1Frame");
}

function activateLayer(name)
{
    var reqParams = "COMMAND=ACTIVATELAYER";
    reqParams += "&SESSION=" + encodeURIComponent(session);
    reqParams += "&MAPNAME=" + encodeURIComponent(mapName);
    reqParams += "&LAYERNAME=" + encodeURIComponent(name);

    if (msie)
        reqHandler = new ActiveXObject("Microsoft.XMLHTTP");
    else
        reqHandler = new XMLHttpRequest();

    reqHandler.open("POST",
"../../moduli/modulo_base/fdo/abfmw_controller.php", false);
    reqHandler.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

    reqHandler.send(reqParams);

    viewerFrame.contentWindow.Refresh();
}

function changeValue(codice)
{
    document.getElementById("valore_cliente").value = codice;
}

```



```

        document.getElementById("form").submit();
    }
</script>
</head>
<body style="padding:0; margin: 0; background: white; background-image:
url('.././../moduli/modulo_base/image/dimming/pf_report_bottom.jpg');
background-repeat: repeat-x; background-position: 0px 150px;">
    <form id="form" action="ricerca_cliente.php" target="" method="POST"
onKeyPress="" style="margin: 0px; margin-top:30px; padding:0px;">
        <input type="hidden" id="valore_cliente" name="valore_cliente"
value="<?php echo $cod_cliente ?>" style="display:none; width:0px"/>
        <table>
            <tr>
                <td width="100px">
                    <label class="reportStile1">Cliente:</label>
                </td>
                <td width="200px">
                    <input type="text" id="cliente" name="cliente" value="<?php echo
$cliente ?>" style="margin-left: 0px; width: 170px; display: inline" />
                    <img id="button_cliente"
src='.././../moduli/modulo_base/image/report/btn_riduci.gif" alt="Lista
Completa"
onmouseover="this.src='.././../moduli/modulo_base/image/report/btn_riduci_over
.gif'; this.style.cursor='hand';"
onmouseout="this.src='.././../moduli/modulo_base/image/report/btn_riduci.gif'"
onclick="document.getElementById('listBlock').style.display=''; getList()"
/>

                    <div id="hint_cliente" class="autocomplete"></div>
                    <script>
                        var params = "";

                        var element_cliente = document.getElementById("valore_cliente");

                        params = "SESSION=" + encodeURIComponent(session) +
"&CHECKSESSION=1";
                        new Ajax.Autocompleter("cliente", "hint_cliente",
"ricerca_cliente_action.php",{afterUpdateElement:getCliente,parameters:params});

                        function getCliente(text, li)
                        {
                            element_cliente.value = li.id;
                            document.getElementById("cliente").value = li.innerHTML;
                            document.getElementById("form").submit();
                        }
                    </script>
                </td>
            </tr>
        </table>
    </form>

```

```

<tr>
  <td height="100" colspan="2" valign="top">
    <div id="listBlock" style="display:none">
      <select style="display: inline; float: left; margin-
right: 5px; margin-left: 5px; width: 85%; height: 95px" name="listBox"
id="listBox" size="5" multiple="multiple"
onchange="changeValue(this.value)">
        </select>
        
      </div>
    </td>
</tr>
<tr align="right">
  <td colspan="2" style="padding-bottom: 10px;">
    <input type="button" value="Cerca" onclick="cerca();
parent.nascondi()" class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
    <input type="button" value="Chiudi" onclick="parent.nascondi()"
class="PulsanteBiancoFix pulsanteGrande"
onMouseOver="this.className='PulsanteOver pulsanteGrande';
style.cursor='hand'" onMouseOut="this.className='PulsanteBiancoFix
pulsanteGrande'" />
  </td>
</tr>
</table>
</form>
</body>
</html>

```

#### **5.6.2.12 ricerca\_cliente\_getList.php**

```

<?php
session_start();
require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'DbHanDBMS.php';
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

$json = new Services_JSON();
$responseType = 'text/plain';
$response = '';

```

```

$dbConnCRM = connectTo('crm',$dbConnSet);

$dbHan = new DbHanDBMS($dbConnCRM, $dbConnSet['crm']['DBMS'],
$args,$args['SESSION']);
$response = $json->encode($dbHan->Select('ID,NAME', 'ACCOUNTS', null, 1,
'NAME,ASC'));
unset($dbHan);

$dbConnCRM -> Close();

header('Content-Type: ' . $responseType);
echo $response;
?>

```

### 5.6.2.13 *ricerca\_cliente\_action.php*

```

<?php
session_start();
$args = ($_SERVER['REQUEST_METHOD'] == "POST") ? $_POST : $_GET;

require_once '../include/siteconstants.php';
require_once '../' . $frameworkPath . 'DbHanFDO.php';
require_once '../' . $frameworkPath . 'DbHanDBMS.php';
session_id($args['SESSION']);

$cliente = $args['cliente'];

$dbConnCRM = connectTo('crm',$dbConnSet);

$dbHan = new DbHanDBMS($dbConnCRM, $dbConnSet['crm']['DBMS'], $args,
$args['SESSION']);
$where = "NAME LIKE '" . $cliente . "%'";
$response = $dbHan->Select('ID,NAME', 'ACCOUNTS', $where, 1, 'NAME,ASC');
unset($dbHan);

$dbConnCRM -> Close();
?>
<ul>
<?php
    if(count($response) != 0)
    {
        $max = count($response);
        if($max > 10)
            $max = 10;
        for($i=0; $i<$max; $i++)
        {
            $sstr = "<li id='" . $response[$i][0] . "'>" . $response[$i][1] .
"</li>";

```

```
        echo $str;
    }
}
?>
</ul>
```

## 6. Risultati raggiunti

### 6.1 Funzionalità di inserimento dati clienti

Durante i test della funzionalità di inserimento dei dati dei clienti è risultato che il tempo impiegato per la geocodifica era elevato: in effetti è stato utilizzato un semplice bubblesort per l'ordinamento delle tratte. Tuttavia da prove effettuate con Google Maps è risultato che il tempo impiegato per la geocodifica è circa lo stesso, quindi questo sta ad indicare che la geocodifica di per sè richiede molto tempo per essere eseguita.

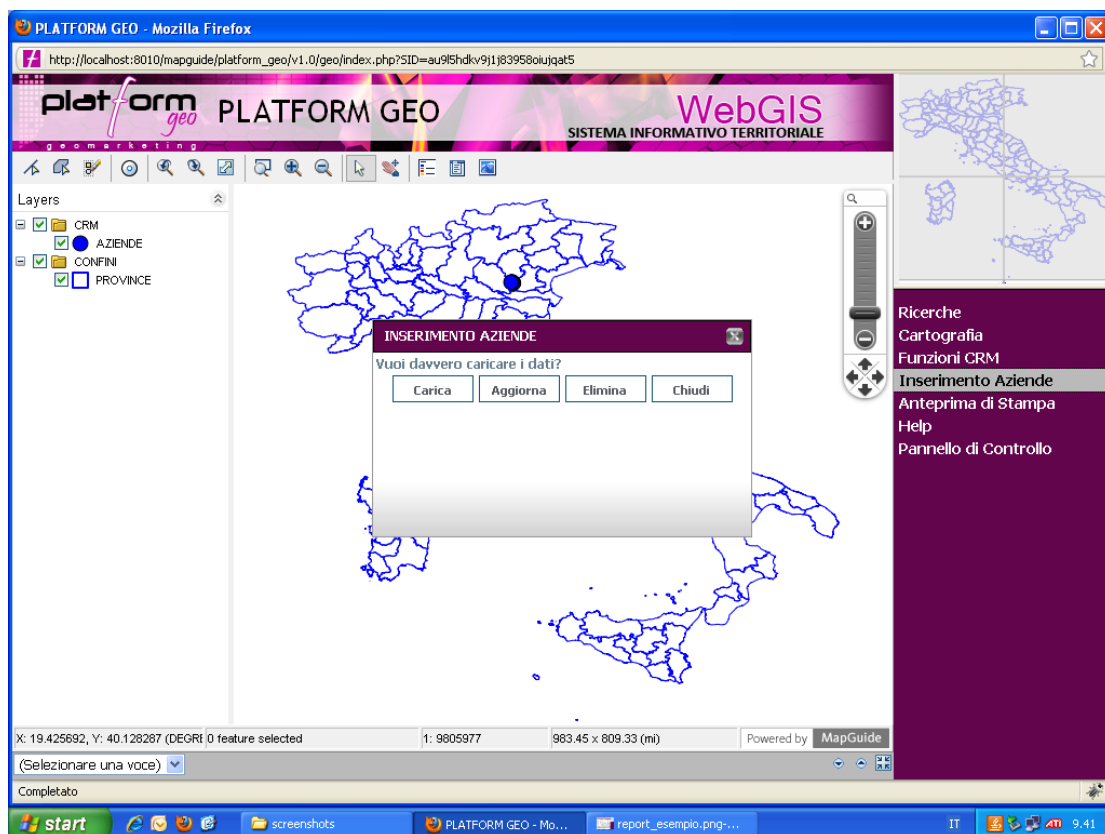


FIGURA 5: INTERFACCIA PER L'INSERIMENTO DEI DATI DEI CLIENTI

Una cosa da valutare è se incorporare la funzione “carica” nella funzione “aggiorna” visto che, per come è stata sviluppata, può essere eseguita soltanto una volta quando i clienti devono ancora essere inseriti.

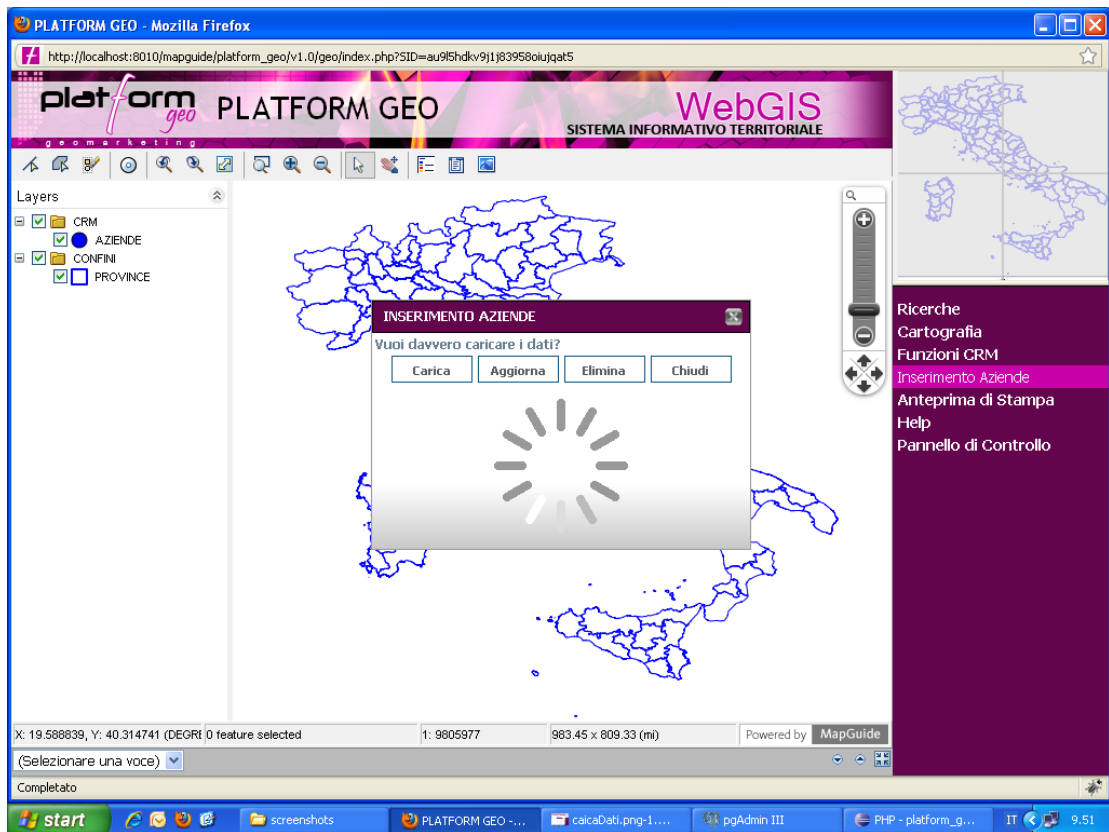


FIGURA 6: CARICAMENTO DEI DATI IN ESECUZIONE

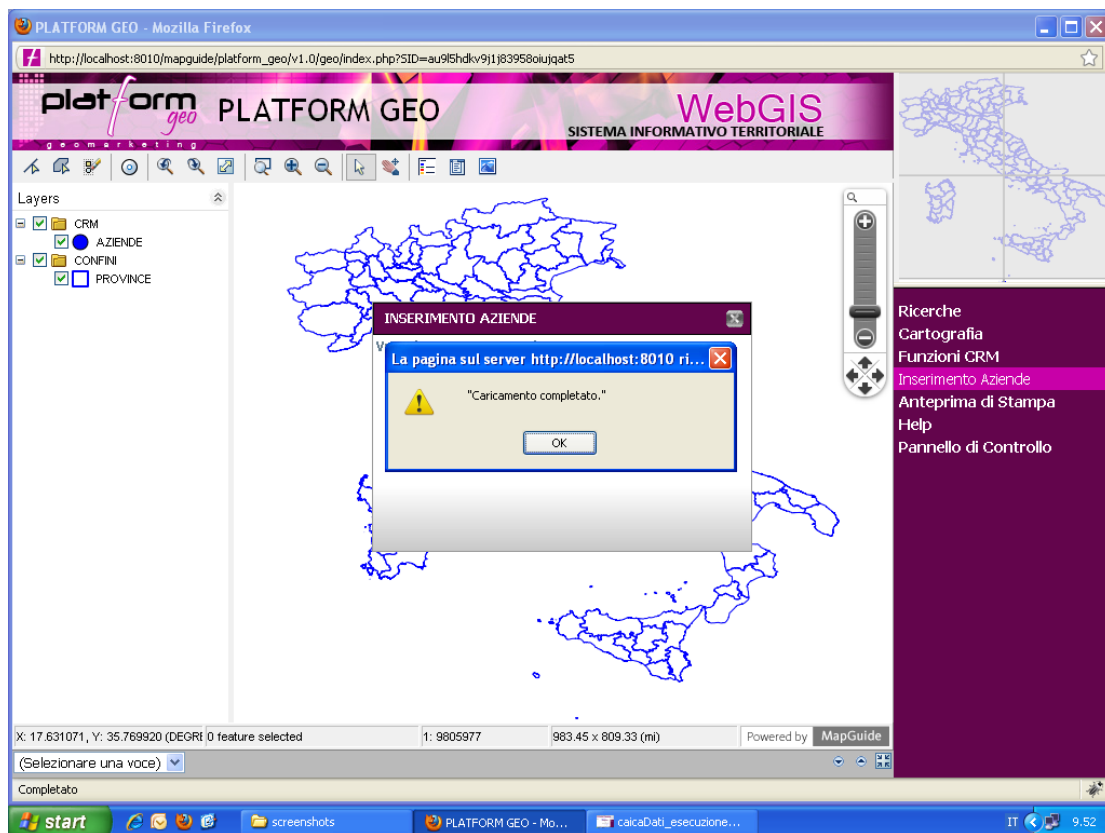


FIGURA 7: CARICAMENTO DEI DATI ESEGUITO

## 6.2 Ricerche

Per quanto riguarda le ricerche non si hanno avuti problemi particolari di prestazioni se non con la ricerca per comune, indirizzo e civico che utilizza una versione in linguaggio Javascript modificata dell'algoritmo di geocodifica. Tale versione realizzata per risparmiare tempo ha violato il requisito di essere indipendente dalla piattaforma utilizzata per la geocodifica. Stessa cosa si può dire per la ricerca per cliente per la quale la lista dei clienti è stata realizzata accedendo direttamente al database del CRM senza utilizzare il componente progettato per lo scopo.

## 6.3 Tematiche statiche per settore, tipo, utente assegnato, categoria

Utilizzando i dati delle aziende memorizzati dall'applicazione sono stati creati dei layer per poter definire delle tematiche statiche che consentono all'utente di prendere decisioni in base ai valori degli attributi utilizzati e alla dislocazione dei clienti sul territorio.

## **7. Sviluppi previsti**

Con questo prototipo sono state implementate le funzioni di base ed evidenziati i problemi a cui si andrà incontro nello sviluppo della versione “ufficiale” del software. Pertanto nelle prossime versioni si cercherà di migliorare gli algoritmi già sviluppati, di sviluppare ciò che in questa versione è stato tralasciato e di realizzare ulteriori funzionalità.

### **7.1 Miglioramenti**

Miglioramento dell’algoritmo utilizzato per ordinare le tratte nell’operazione di geocodifica

Miglioramento delle ricerche per comune indirizzo e civico e quella per cliente

### **7.2 Nuove funzionalità**

#### **7.2.1 Filtri dinamici**

Si prevede lo sviluppo di filtri dinamici ovvero di filtri che l’utente potrà definire al momento per ottimizzare le decisioni da prendere.

#### **7.2.2 Inserimento di ulteriori dati**

Nelle prossime versioni verranno inseriti anche i dati relativi alle offerte, alla concorrenza e alle campagne marketing sempre con lo scopo di permettere di effettuare decisioni opportune