



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

**PROGETTAZIONE E SVILUPPO DI  
UN'APPLICAZIONE WEB ORIENTATA AL CRM**

Laureando  
Luca Martini

Relatore  
Marcello Dalpasso

Anno accademico 2011/2012



## **SOMMARIO**

Questa tesi vuole illustrare il lavoro svolto durante il tirocinio presso l'azienda Techgest s.a.s. di Trevignano (TV) nei mesi di marzo, aprile, maggio e giugno 2011. La finalità di questo tirocinio è stata quella di collaborare allo sviluppo e al testing di un'applicazione web dedicata alla gestione dei clienti, in particolare la gestione dei contatti e delle pratiche ad essi relative.

Tale software è stato commissionato inizialmente da uno studio commercialistico, ma l'obiettivo è quello di creare una base generica con alcune funzioni principali, alla quale poter aggiungere caratteristiche aggiuntive in base alle esigenze specifiche del cliente richiedente.

L'azienda ha deciso di sviluppare questo software tramite lo strumento dell'applicazione web, per via di una serie di vantaggi che esso comporta, tra cui la compatibilità con la maggior parte dei sistemi informatici e la semplicità di utilizzo per l'utente.

Data l'esperienza già accumulata precedentemente dai dipendenti dell'azienda con gli ambienti Microsoft, si è scelto di utilizzare gli strumenti di sviluppo della piattaforma .NET, che comprendono il linguaggio ASP .NET, il linguaggio C# e il DBMS MS SQL Server. Questi strumenti sono stati facilmente gestiti attraverso le suite Microsoft Visual Studio 2010 e Microsoft SQL Server 2008 R2.



## INDICE

1. Introduzione	pag. 1
2. Il concetto di CRM	pag. 4
3. Le applicazioni web	pag. 6
3.1. La storia	pag. 7
3.2. I vantaggi	pag. 9
4. Gli strumenti utilizzati	pag. 10
4.1. La piattaforma Microsoft .NET	pag. 11
4.1.1. Il Framework .NET	pag. 13
4.1.1.1. La versione 4	pag. 19
4.1.2. IIS	pag. 20
4.2. Microsoft Visual Studio	pag. 21
4.3. Tortoise SVN	pag. 23
5. La struttura di un'applicazione web	pag. 25
5.1. La visualizzazione tramite browser	pag. 26
5.1.1. Il linguaggio ASP .NET	pag. 27
5.2. Il motore applicativo	pag. 29
5.2.1. Il linguaggio C#	pag. 30
5.3. Il database	pag. 32
5.3.1. Il DBMS MS SQL	pag. 34
6. L'applicazione	pag. 35
6.1. Gli obiettivi	pag. 36
6.2. Il login	pag. 37
6.3. La pagina iniziale	pag. 38
6.4. Le anagrafiche	pag. 41
6.4.1. I clienti	pag. 42
6.4.2. I contatti	pag. 45
6.5. Le note	pag. 47
6.6. La gestione	pag. 49
6.6.1. Le pratiche	pag. 50
6.6.2. Gli adempimenti	pag. 52
6.6.3. Le prestazioni	pag. 54
6.7. L'amministrazione	pag. 56
7. Conclusioni	pag. 58



## **1 INTRODUZIONE**

Grazie all'aumento della potenza delle connessioni internet, sia a livello professionale che a livello domestico, in questi ultimi anni si è via via diffuso l'utilizzo delle applicazioni web, molto spesso preferite al posto dei tradizionali software eseguibili offline. Si parla quindi di siti web talmente interattivi e con funzionalità così avanzate, da essere considerati delle vere e proprie applicazioni, fruibili online o attraverso una rete interna. Il vantaggio principale di questo approccio è ovviamente la possibilità di eseguire tutte le operazioni attraverso un normale browser, senza quindi bisogno di lunghe e talvolta difficoltose installazioni su macchine dalle diverse caratteristiche e quindi con probabili problemi di compatibilità. Anche l'aggiornamento avrà un costo ridotto, essendo centralizzato e non diffuso su ogni terminale. Per non parlare poi della possibilità di usufruire di tali applicazioni anche in situazioni di mobilità, attraverso notebook, palmari o addirittura smartphone.

L'azienda dove si è svolto il tirocinio, in passato aveva già sviluppato con questo stesso strumento un software per il CRM (Customer Relationship Management ovvero gestione delle relazioni con i clienti) con poche e semplici funzioni a disposizione. Per lo più si trattava di gestire un elenco di clienti, ognuno associato a diversi tipi di contatto (indirizzo, telefono, email, ecc.). Su richiesta di uno studio commercialistico si è poi deciso di realizzare una seconda versione di questo software più potente e completa, cercando di soddisfare tutte le esigenze del cliente ma mantenendo un'interfaccia abbastanza semplice e soprattutto intuitiva. Uno degli obiettivi infatti è stato quello di rendere questo applicativo di facile utilizzo così da ridurre al minimo la formazione necessaria al personale che lo utilizzerà. Ovviamente, anche se il software è stato commissionato da un cliente ben preciso, si è cercato di sviluppare una base con strumenti generici che possa essere poi personalizzata a seconda del richiedente.

Fortunatamente in commercio esistono già degli strumenti molto potenti a disposizione degli sviluppatori, e per via di una conoscenza già consolidata dell'ambiente di sviluppo Microsoft, si è scelto di optare per gli strumenti della piattaforma .NET, probabilmente tra le più utilizzate attualmente.

Gli strumenti che si è deciso di utilizzare sono i seguenti :

- Il linguaggio ASP .NET
- Il linguaggio di alto livello C#
- Il DBMS MS SQL Server

Le pagine che vengono visualizzate dal browser sono scritte in linguaggio ASP .NET, un linguaggio visivo con modalità di funzionamento simili all'HTML, anche se ovviamente più complesso, che richiama funzioni del programma scritte in un linguaggio di alto livello, ovvero il C#. Esso offre molte possibilità per inserire all'interno delle pagine web dei controlli avanzati per realizzare un'interfaccia molto avanzata graficamente.

Se si utilizza il framework .NET, ovvero l'ambiente di sviluppo per tutti gli applicativi .NET, è possibile scrivere in molti linguaggi di programmazione di alto livello tra cui :

- C#
- Visual Basic .NET
- J#
- Perl
- Python

Il codice scritto tramite uno di questi linguaggi viene tradotto in un linguaggio intermedio comune (CIL ovvero Common Intermediate Language) che viene poi compilato ed eseguito dalla macchina virtuale (CLR ovvero Common Language Runtime).



Il linguaggio di programmazione di alto livello C# (si legge "C sharp") è un linguaggio object-oriented la cui sintassi prende spunto da quella del Delphi, del C++, di Java e di Visual Basic. Questo linguaggio è stato creato da Microsoft specificatamente per la programmazione nel framework .NET ed ha quindi con esso una stretta integrazione. I suoi tipi di dati primitivi hanno una corrispondenza univoca con i tipi .NET e molte delle sue astrazioni come classi, interfacce, delegati ed eccezioni sono particolarmente adatte a gestire il già citato framework.

Il DBMS relazionale Microsoft SQL Server è prodotto da Microsoft, inizialmente veniva utilizzato per basi di dati medio-piccole ma dal 2000 è stato potenziato anche per quelle di grosse dimensioni. Esso utilizza una variante del linguaggio SQL standard chiamata T-SQL (Transact-SQL).

L'IDE (Integrated Development Environment ovvero ambiente di sviluppo integrato) che è stato utilizzato in questo progetto è Microsoft Visual Studio 2010, il quale permette la realizzazione di applicazioni, siti web, applicazioni web e servizi web. E' inoltre un RAD (Rapid Application Development) ovvero un'applicazione atta ad aumentare la produttività aiutando il programmatore con mezzi come l'IntelliSense (uno strumento di completamento automatico delle funzioni) o un designer visuale delle forms.

Al momento del mio arrivo nell'azienda come tirocinante era già stata sviluppata una release iniziale dell'applicazione web. Inizialmente quindi il mio scopo è stato quello di testare questa versione iniziale dell'applicazione verificando tutte le caratteristiche alla ricerca di eventuali bug, lacune o anche semplici errori grafici dell'interfaccia. Successivamente, una volta aumentata la confidenza con gli strumenti di sviluppo precedentemente citati, è iniziata la collaborazione allo sviluppo di ulteriori funzionalità richieste dal cliente, le quali venivano poi verificate man mano tramite continui colloqui con esso.

## 2 IL CONCETTO DI CRM

Il concetto di Customer Relationship Management (spesso abbreviato in CRM) oppure gestione delle relazioni coi clienti è legato al concetto di fidelizzazione dei clienti.

In un'impresa il mercato non è rappresentato esclusivamente dal cliente inteso come persona fisica ma anche dall'ambiente circostante, con il quale l'impresa deve stabilire relazioni durevoli (sia di breve che di lungo periodo). E' quindi fondamentale pianificare ed implementare apposite strategie.

Il CRM si propone principalmente quattro obiettivi :

- L'acquisizione di nuovi clienti o di potenziali clienti.
- L'aumento delle relazioni con i clienti più importanti.
- La fidelizzazione il più longeva possibile dei clienti che hanno maggiori rapporti con l'impresa.
- La trasformazione degli attuali utenti in procuratori ovvero di consumatori che lodano l'azienda incoraggiando altre persone a rivolgersi alla stessa.

Per raggiungere questi scopi vi sono essenzialmente tre tipi di CRM :

- CRM operativo : soluzioni metodologiche e tecnologiche per automatizzare i processi a contatto diretto con il cliente.
- CRM analitico : procedure e strumenti per migliorare la conoscenza del cliente tramite l'estrazione di dati dal CRM operativo, la loro analisi e lo studio sui comportamenti dei clienti stessi.
- CRM collaborativo : metodologie e tecnologie integrate con gli strumenti di comunicazioni per gestire i contatti con il cliente.

Sarebbe riduttivo equiparare il concetto di CRM a quello di un semplice software. Il CRM non è semplice marketing o informatica, bensì una strategia di comunicazione e di integrazione tra processi aziendali che pone il cliente al centro dell'attenzione.

Un'azienda che intende seguire la strada del CRM dev'essere consapevole che prima di investire nella tecnologia deve comunque investire nell'organizzazione interna. Inoltre il CRM è più orientato alle aziende che cercano un ritorno di investimento nel tempo piuttosto che veloce.

Quelle che possono essere considerate le componenti strutturali di un CRM sono :

- Analisi e gestione della relazione con i clienti : contatto con i clienti e analisi dei bisogni attraverso una molteplicità di strumenti. Il contatto è fondamentale se si vuole mappare ogni singolo cliente per poi organizzare tutte le informazioni raccolte in un database strutturato. Queste informazioni sono preziose in quanto permettono di conoscere e se possibile anticipare le esigenze del cliente.
- Lo sviluppo di contenuti e servizi personalizzati : i dati raccolti vengono gestiti per elaborazioni statistiche utili a segmentare i clienti in specifiche scale. Una volta organizzati è possibile procedere con l'analisi per sviluppare un'offerta commerciale personalizzata.
- L'infrastruttura informatica : l'attivazione di strumenti informatici che aiutano in questo processo di gestione del cliente.

Un ottimo sistema CRM comprende una serie di infrastrutture sia a livello di relazioni con l'esterno che a livello di organizzazione interna.

Le applicazioni orientate al CRM sono atte alla semplificazione dei contatti con i clienti permettendo il salvataggio delle loro informazioni in un database. Esse forniscono le modalità per interagire in modo che tali interazioni possano essere registrate e analizzate.

### **3 LE APPLICAZIONI WEB**

“Applicazione web” è l’espressione che in generale si usa per indicare tutte le applicazioni distribuite web based. Per essere precisi, questa espressione è impiegata nell’ambito del software engineering, dove con la definizione di applicazione web si descrive un’applicazione accessibile via web per mezzo di un network, come ad esempio una intranet, oppure attraverso la rete Internet.

Come già espresso nella precedente introduzione, le applicazioni web stanno conoscendo in questo momento una diffusione come mai era successo in passato grazie a molteplici fattori. Primo fra tutti il costante aumento di velocità delle connessioni. Sono infatti sempre più veloci le reti locali (LAN) ma anche quelle con accesso a internet, sia se si parla di un’utenza di tipo domestico sia se si parla di un’utenza di tipo aziendale. L’introduzione e la diffusione della banda larga hanno comportato tempi di risposta sempre più brevi nella navigazione, rendendo immediate quelle operazioni che prima richiedevano lunghe attese o addirittura rischiavano di non andare a buon fine a causa del superamento dei tempi di timeout.

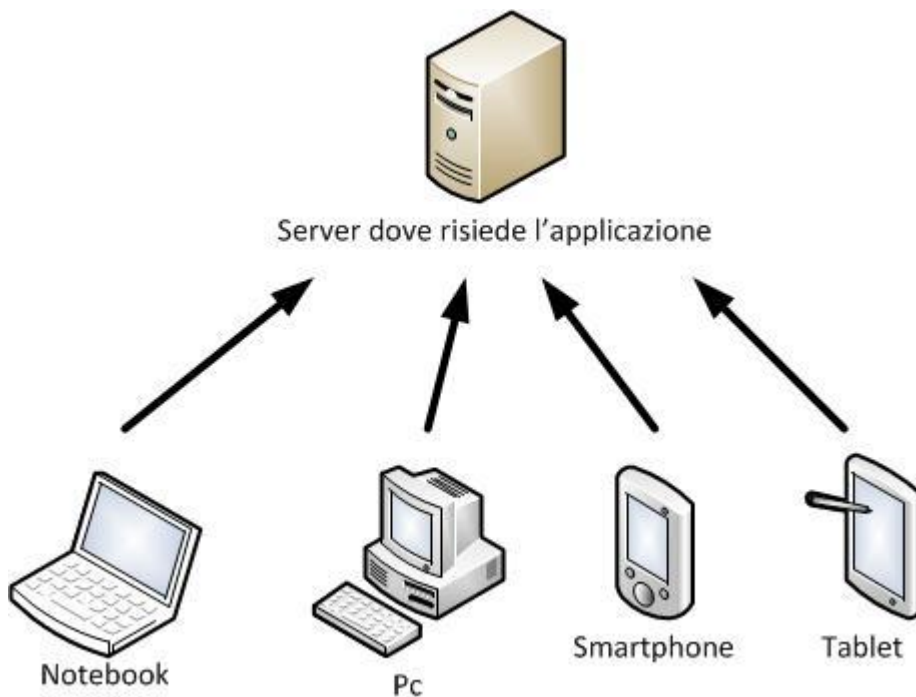
Nella pratica i siti web si sono evoluti verso una dimensione sempre più interattiva con l’utente e si è passati da una semplice visualizzazione di informazioni ad una elaborazione di dati in tempo reale vera e propria. E’ da rilevare che recentemente alcune grandi realtà informatiche hanno contribuito nel diffondere le applicazioni web, implementando interi pacchetti applicativi online, anzi a volte facendone una vera e propria strategia aziendale, come nel caso di Google.

### **3.1 LA STORIA**

Quando verso circa metà degli anni ottanta cominciavano ad essere rilasciate le prime applicazioni client-server, questi sistemi richiedevano che per ognuno di essi fosse installata la parte client direttamente sulla macchina di ciascun utente. Di fatto qualunque aggiornamento o modifica dell'applicativo lato server si trasformava nell'esigenza di installare versioni aggiornate su tutte le postazioni degli utenti. Questo tipo di problematiche induceva ovviamente ad una spesa economica non indifferente per le aziende che decidevano di utilizzare il software.

Al contrario le web application generano il loro output in modo dinamico pilotando un frontend composto da documenti web, destinati quindi ad un semplice browser. Le prime webapps consistevano infatti nella generazione di pagine standard HTML o XHTML; successivamente con l'evolversi delle tecnologie associate e con la nascita di nuovi standard, si cominciarono a distribuire attraverso di esse documenti in formati più avanzati ed ottimizzati, come ad esempio l'XML. In breve, ogni singola pagina web viene inviata al client come fosse un documento statico, ma la sequenza delle pagine fornisce un'esperienza interattiva. Durante la sessione il browser interpreta e visualizza le pagine e dunque diventa il client universale per tutte le web application, indipendentemente dal produttore del software. Il crescente successo conseguito da librerie esperte, ormai veri e propri framework come JQuery, tecnologie dinamiche come AJAX, oppure plugin, come il conosciutissimo Flash Player, consente di arricchire le interfacce utente in modo completo ed avanzato.

Bisogna poi tener conto della velocissima diffusione in ambito lavorativo degli smartphone. Telefonini con un proprio sistema operativo talmente evoluti da possedere un browser con potenzialità pari a quelle di un comune browser per pc. Essi diventano strumenti importantissimi per poter accedere, molto spesso sfruttandone tutte le funzionalità, alle applicazioni web. In questo modo l'utente può aggiungere o modificare dati senza tornare fisicamente in azienda, e non è un vantaggio di poco conto per quanto riguarda il risparmio sia economico che temporale. Insieme agli smartphone si stanno via via diffondendo anche altri tipi di dispositivi mobili con la disponibilità di un accesso a internet come ad esempio i tablet. Di conseguenza si stanno moltiplicando anche gli strumenti per accedere ad internet in situazioni di mobilità come le reti wifi o le chiavette degli operatori mobili per la navigazione attraverso la rete dei cellulari.



## **3.2 I VANTAGGI**

Come abbiamo già accennato precedentemente, tutti i vantaggi principali girano intorno all'utilizzo del browser come client per eseguire un'applicazione web. Questo prima di tutto fa sì che non ci siano problemi di compatibilità ad esempio con sistemi operativi diversi, sia se si parla di personal computer che di smartphone o di altri supporti elettronici. Non ci sono quindi spese da parte dell'azienda per adeguare i sistemi operativi già presenti sulle proprie macchine.

Alla quasi totale assenza di requisiti minimi software dobbiamo aggiungere anche l'assenza di requisiti minimi hardware. Non è infatti necessario ricorrere ai sistemi più recenti, è sufficiente che siano in grado di far eseguire un normalissimo browser. Ovviamente in configurazioni più recenti i tempi di esecuzione saranno più brevi, ma normalmente la differenza non è così considerevole se paragonati a sistemi più obsoleti.

Continuando con i vantaggi, si può notare l'assenza di una vera e propria installazione, se escludiamo quella appunto del browser, normalmente già presente di default in tutti i sistemi operativi. Questo sicuramente comporta un notevole risparmio di tempo che aumenta ancora di più se pensiamo anche all'assenza di tempi di aggiornamento, o meglio di tempi di aggiornamento per le singole postazioni dell'utente. Se infatti devono essere apportate delle modifiche al software, esse saranno introdotte in modo centralizzato, quindi con un singolo intervento, e saranno da subito operative per tutte le postazioni. Si può immaginare quale passo avanti significativo sia quest'ultimo, in particolar modo per aziende dotate di sistemi informatici di grandi dimensioni.

Infine, dato che solitamente un utente medio è in grado senza problemi di navigare in internet (o almeno lo si spera) sarà più facile per il personale l'utilizzo di un'applicazione web. Sicuramente ci sarà bisogno di imparare alcune funzionalità, soprattutto quelle più avanzate, ma l'approccio via browser è sicuramente un aiuto in termini di intuitività, essendo un ambiente già familiare a chi normalmente utilizza uno strumento informatico.

## **4 GLI STRUMENTI UTILIZZATI**

L'azienda ha deciso di utilizzare gli strumenti messi a disposizione dalla piattaforma .NET di Microsoft, data l'esperienza già accumulata in passato con queste tecnologie dai dipendenti.

La piattaforma Microsoft .NET mette a disposizione una vasta gamma di strumenti molto potenti per la progettazione e la realizzazione di applicazioni web, progettati in modo da funzionare in modo integrato all'interno della piattaforma stessa.

Per una gestione di tutti questi strumenti la scelta software più logica ricade senza dubbio sulla suite di sviluppo Microsoft Visual Studio. Esso è infatti un ambiente di sviluppo interamente concepito proprio per l'ambiente .NET. L'azienda ha scelto di utilizzare la versione 2010 della suite.

Per garantire ad ogni dipendente un lavoro autonomo senza generare conflitti causati dall'utilizzo in contemporanea degli stessi file si è scelto di utilizzare il client Subversion gratuito TortoiseSVN, il quale si integra perfettamente con il sistema operativo Windows. Tramite questo client è possibile indicare quando un dipendente sta lavorando su un determinato file e risalire alle versioni precedenti dello stesso.



## 4.1 LA PIATTAFORMA MICROSOFT .NET



Microsoft .NET (si pronuncia “dotnet”) è un progetto all’interno del quale Microsoft ha creato una piattaforma di sviluppo software che utilizza una versatile tecnologia di programmazione ad oggetti.

.NET nasce come contrapposizione al linguaggio Java ed al contrario di esso è uno standard ISO riconosciuto, non è quindi possibile modificarne la sintassi da parte della casa madre. La prima versione è stata rilasciata nel 2002, come evoluzione del linguaggio J++, ovvero una versione evoluta di Java realizzata da Microsoft nel 1998.

Una caratteristica primaria di .NET è l’indipendenza dalla versione di Windows da cui è installata. Inoltre, anche se Microsoft non ha implementato il framework .NET per nessuna piattaforma diversa da Windows, ha reso disponibili le specifiche affinché sia possibile la sua implementazione su altri sistemi operativi.

L’infrastruttura utilizza in modo esteso il concetto di modularità dei componenti COP (Component Oriented Programming) e diventa di fatto un’evoluzione dell’esistente modello COM (Component Object Model). Essa include molte funzionalità progettate espressamente per integrarsi in ambiente internet.

Il già citato CLR (Common Language Runtime) supervisiona l’esecuzione di tutti i programmi garantendo tra l’altro la gestione della memoria, della sicurezza e delle eccezioni. Il codice scritto nei vari linguaggi viene tradotto in un linguaggio intermedio IL (Intermediate Language) che viene poi eseguito da questo strumento.

La presenza del CTS (Common Type System) definisce tutti i diversi tipi e costrutti che il CLR supporta e le modalità per la creazione di nuovi tipi. In questo modo .NET supporta lo scambio di istanze tra librerie scritte utilizzando linguaggi differenti.

La disponibilità della Base Class Library fornisce classi che implementano funzionalità di base riguardanti la gestione dei file, la manipolazione grafica, l'interazione con i database e molti altri strumenti.

Grazie a degli specifici namespaces presenti nella piattaforma è possibile accedere a funzionalità implementate in applicazioni esterne all'ambiente .NET. Questo garantisce una compatibilità completa tra nuove applicazioni e quelle meno recenti.

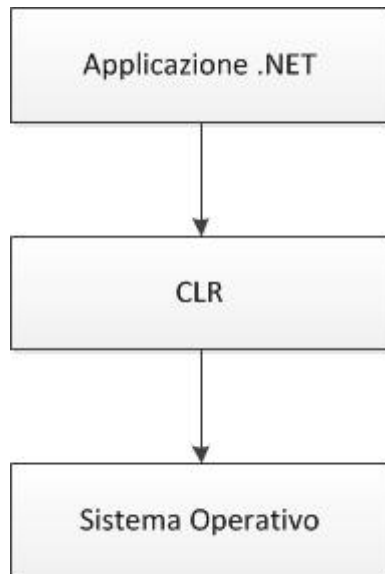
La sicurezza è garantita da alcuni strumenti atti ad affrontare alcune delle vulnerabilità più comuni che vengono sfruttate per scrivere del software dannoso. E' stato poi introdotto un modello comune di sicurezza per tutte le applicazioni.

### 4.1.1 IL FRAMEWORK .NET

Il framework .NET si occupa di ogni aspetto riguardante lo sviluppo dell'applicazione, a prescindere dal sistema operativo sottostante. Esso offre al programmatore un ambiente di sviluppo completo e si occupa di ogni aspetto come ad esempio la gestione delle risorse e della memoria, l'accesso ai dati, la gestione dell'interfaccia e così via. Il framework si divide in tre parti :

- Compilatori per i principali linguaggi supportati
- Ambiente di esecuzione CLR (Common Language Runtime)
- Libreria di classi

Il componente principale fra questi è senza dubbio il CLR : esso è un livello del framework posto al di sopra del sistema operativo, il quale si occupa dell'esecuzione delle applicazioni. Queste ultime infatti non comunicano direttamente con il sistema operativo ma lo fanno tramite il CLR che gestisce il rispetto delle dipendenze, la gestione della memoria, la sicurezza, l'integrazione del linguaggio e molto altro ancora. Inoltre fornisce molti servizi che facilitano la scrittura del codice e la distribuzione dell'applicazione oltre a migliorarne l'affidabilità. Il codice al quale si riferisce è detto "codice gestito" (managed code) mentre il codice generato dai compilatori viene detto "codice non gestito" (unmanaged code) perché scavalca il CLR. A questo componente viene anche demandata la responsabilità per attività quali la creazione di oggetti e l'esecuzione di chiamate a metodi.



A sua volta, il Common Language Runtime è diviso in cinque componenti, ovvero :

- CTS (Common Type System) : è il sistema di tipi di dato unificato. Oltre a definire uno standard di tipi di dato pone anche le regole necessarie alla realizzazione di nuovi tipi. In questo modo il CLR è in grado di capire come crearli ed eseguirli. Sarà quindi possibile per i compilatori del framework .NET utilizzare i servizi del CLR per definire questi nuovi tipi di dato, gestire gli oggetti ed eseguire chiamate a metodi, invece di utilizzare i metodi specifici appartenenti al linguaggio di programmazione utilizzato. L'utilizzo di questo sistema ha come conseguenza una profonda integrazione tra diversi linguaggi di programmazione. Basti pensare che il codice scritto in un linguaggio può ereditare l'implementazione da classi scritte in un altro; oppure che le eccezioni sollevate dal codice scritto in un linguaggio possono anch'esse essere gestite da codice scritto in un altro linguaggio. Sicuramente questo comporta dei vantaggi notevoli per gli sviluppatori, i quali non necessitano più di creare differenti versioni delle proprie librerie a seconda del linguaggio di programmazione. Un risparmio considerevole di tempo e lavoro di programmazione.

- CLS (Common Language Specification) : è una serie di regole che vengono applicate per generare gli assembly. Se un componente scritto in un linguaggio deve essere utilizzato da un altro linguaggio allora lo sviluppatore dovrà aderire ai tipi e alle strutture definite dal CLS. Alcune regole importanti che vengono definite sono :
  - evitare l'utilizzo di nomi che corrispondono comunemente a parole chiave nei linguaggi di programmazione;
  - non creare tipi di dato nidificati;
  - assumere che le implementazioni dei metodi con lo stesso nome in interfacce differenti siano indipendenti.

Un linguaggio progettato per accedere a tutte le caratteristiche fornite dal CLS è detto "CLS Consumer". Esso deve poter svolgere determinate operazioni, tra le quali :

- supportare chiamate da ogni metodo conforme al CLS;
- possedere un meccanismo per chiamare i metodi con i nomi che sono parole chiave nel linguaggio;
- creare un'istanza di ogni tipo di dato conforme al CLS;
- leggere e modificare campi conformi al CLS;
- accedere a tipi di dato nidificati;
- accedere a qualunque proprietà ed evento aderente al CLS.

Un superset del CLS Consumer che offre al programmatore totale libertà di azione è detto "CLS Extender".

- CIL (Common Intermediate Language) : è una rappresentazione intermedia del codice, indipendente dalla cpu, che deve essere generata da qualsiasi compilatore che aderisce alla struttura del CLR. L'implementazione del CIL nel framework è chiamata Intermediate Language (IL). Il linguaggio intermedio IL si pone ad un livello più elevato dei linguaggi macchina ordinari, avendo al suo interno istruzioni per il caricamento, la memorizzazione e l'inizializzazione dei dati, per richiamare dei metodi, per eseguire operazioni aritmetiche e logiche, per controllare il flusso e l'accesso alla memoria e per gestire gli errori e le eccezioni. Questo formato intermedio ha alcune caratteristiche in comune con il tradizionale linguaggio Assembly, come ad esempio la possibilità di operare sui dati con istruzioni del tipo "push" e "pop" oppure la capacità di spostarli in dei registri. Diversamente da esso però non fa riferimento a nessuna specifica piattaforma hardware. Un grande vantaggio dell'usare questa soluzione è la possibilità da parte del CLR di verificare durante la compilazione che il codice sia conforme alle specifiche di programmazione ed ai tipi di dato previsti. Ad esempio durante questa verifica viene controllato il corretto uso dei puntatori o la correttezza delle conversioni. Analogamente a Java, prima dell'esecuzione il linguaggio intermedio CIL viene convertito al volo dal JIT (vedi il punto seguente) in codice specifico per la cpu. Quando un compilatore conforme al CLS genera linguaggio intermedio produce anche dei metadati che descrivono i tipi di dato specifici utilizzati nel codice. L'implementazione MSIL e i metadati sono contenuti in file Portable Executable (PE) che all'utente appaiono come familiari file con estensione .DLL e .EXE. La presenza dei metadati in questi file permette la descrizione dei componenti eliminando di fatto la necessità di librerie aggiuntive dato che il CLR localizza e legge i metadati ove necessario durante l'esecuzione del codice.

- JIT (Just In Time Compiler) : è un compilatore che converte il codice CIL in codice nativo specifico della cpu sulla quale sta funzionando. Il principio di funzionamento si basa sul fatto che alcune parti di codice di un programma potrebbero non essere mai chiamate in causa durante l'esecuzione del programma, quindi per non sprecare tempo e memoria nella conversione di tutto il linguaggio intermedio in codice nativo, il JIT converte il CIL al bisogno durante l'esecuzione. Nello specifico, il loader crea e allega una "stub" (componente software) ad ogni metodo, quando quest'ultimo viene chiamato, lo stub passa il controllo al compilatore JIT il quale converte il codice CIL di quel metodo in codice nativo e modifica lo stub affinché diriga l'esecuzione alla locazione del codice nativo. Ciò significa che le chiamate successive di quel metodo verranno dirette verso il codice nativo generato precedentemente, riducendo il tempo necessario alle successive esecuzioni del programma. Se un metodo non viene utilizzato per un determinato periodo di tempo il runtime preleva il blocco di codice compilato e lo rimuove dalla memoria, sostituendolo con uno stub che permette di individuarlo nuovamente al fine di rigenerare il codice nativo completo. Il codice così compilato (sia mediante JIT che direttamente in forma nativa al momento dell'installazione) deve sottoporsi ad un processo di verifica che esamina il codice CIL e i metadati per stabilire che siano "Type Safe" ovvero che accedano esclusivamente alle locazioni di memoria autorizzate, che le identità siano verificate e che i riferimenti ai tipi di dati siano compatibili con i tipi stessi. Questa caratteristica offre uno strato di protezione automatico dagli errori di programmazione. Probabilmente ad un confronto diretto il codice compilato dal JIT risulterà comunque più lento del tradizionale codice non gestito, ad esempio proveniente da un compilatore C/C++, tuttavia per sua stessa natura il compilatore JIT lavora direttamente sulla macchina di destinazione del programma e pochi istanti prima di eseguire il programma stesso. Questa importante caratteristica fornisce al JIT dati sull'ambiente di destinazione che nessun compilatore tradizionale potrà mai avere, ottenendo così un

elevato grado di ottimizzazione. Ad esempio il compilatore JIT può rilevare che la piattaforma dispone di un processore Intel Core i5 e compilare il codice con le specifiche istruzioni CPU, mentre con una procedura tradizionale questa scelta viene compiuta in fase di sviluppo del software. Anche l'esatta conoscenza dello stato della memoria e dei registri può rappresentare un dato significativo per ottimizzare ulteriormente il codice. Sono previsti due tipi di compilatori JIT : una versione standard e una versione ridotta "economy". Il compilatore standard esamina il codice CIL di un metodo e lo converte in codice nativo ottimizzato per la piattaforma esattamente come fa un tradizionale compilatore C/C++. La versione economy invece è stata progettata per l'utilizzo su macchine per le quali il costo per l'uso della memoria e della cpu è elevato, come ad esempio sistemi basati su Windows CE. Il compilatore economy semplicemente non fa altro che rimpiazzare ogni istruzione IL con la sua controparte nativa. Come si può immaginare questo tipo di compilazione è molto più rapida di quella standard, tuttavia il codice prodotto è molto meno efficiente in quanto non soggetto all'ottimizzazione per la piattaforma specifica. Nonostante questo, il codice risulta comunque più efficiente di normale codice interpretato.

- VES (Virtual Execution System) : è l'equivalente della Java Virtual Machine in ambiente Sun. Il VES carica ed esegue i programmi scritti per il CLR. Esso adempie alle sue funzioni utilizzando le informazioni contenute nei metadati, fornisce inoltre dei servizi durante l'esecuzione dei codici, ad esempio la gestione automatica della memoria oppure il debugging.



#### 4.1.1.1 LA VERSIONE 4

La prima beta del framework fu rilasciata verso la fine del 2000. Il framework è stato quindi aggiornato più volte nel corso degli anni.

VERSIONE	DATA DI RILASCIO
1.0	05/01/2002
1.1	01/04/2004
2.0	07/11/2005
3.0	06/11/2006
3.5	19/11/2007
4.0	12/04/2010

Il framework .NET 4.0 è stato rilasciato con l'ambiente di sviluppo Microsoft Visual Studio 2010 il 12 aprile 2010. Le novità di questa versione includono miglioramenti ai seguenti aspetti :

- modello di sicurezza
- distribuzione e compatibilità delle applicazioni
- nuove funzionalità principali
- libreria Managed Extensibility Framework
- elaborazione parallela
- servizi di rete
- web
- client
- Windows Communication Foundation
- Windows Workflow Foundation
- nuovi tipi di dato e membri

## 4.1.2 IIS

IIS, ovvero Microsoft Internet Information Service, è un insieme di servizi server Internet per sistemi operativi Microsoft Windows. All'inizio è stato distribuito opzionalmente in aggiunta al sistema operativo Windows NT, da Windows 2000 in poi invece è stato integrato all'interno del sistema operativo. Attualmente è arrivato alla versione 7.5 ed include servizi server per i protocolli FTP, SMTP, NNTP e HTTP/HTTPS.

<b>VERSIONE IIS</b>	<b>VERSIONE WINDOWS</b>
1.0	Windows NT 3.51
2.0	Windows NT 4.0
3.0	Windows NT 4.0 Service Pack 3
4.0	Windows NT 4.0 Option Pack
5.0	Windows 2000
5.1	Windows XP Professional
6.0	Windows Server 2003
7.0	Windows Vista e Windows Server 2008
7.5	Windows 7

L'applicazione in sé non è in grado di eseguire elaborazioni "server-side" ma ne delega l'esecuzione ad applicazioni ISAPI. ISAPI (Internet Server Application Programming Interface) è un'interfaccia di programmazione (API) che Microsoft ha rilasciato appositamente per gli sviluppatori che vogliono ampliare gli strumenti di IIS. Inoltre Microsoft fornisce una serie di applicazioni tra le quali il modulo per ASP e ASP .NET.

Per IIS negli anni passati sono stati segnalati numerosi problemi di sicurezza, i quali sono stati per la maggior parte risolti con le ultime versioni rilasciate.

## 4.2 MICROSOFT VISUAL STUDIO



Visual Studio è un IDE (Integrated Development Environment ovvero ambiente di studio integrato) sviluppato da Microsoft, che supporta diversi tipi di linguaggio ad alto livello e che permette la realizzazione di applicazioni web. Inoltre esso è un RAD (Rapid Application Development) ovvero un'applicazione preposta ad aumentare la produttività del programmatore aiutandolo con diversi strumenti. Esso è multiplatforma : permette infatti la realizzazione di programmi per server, workstation, pocket PC, smartphone e ovviamente browser.

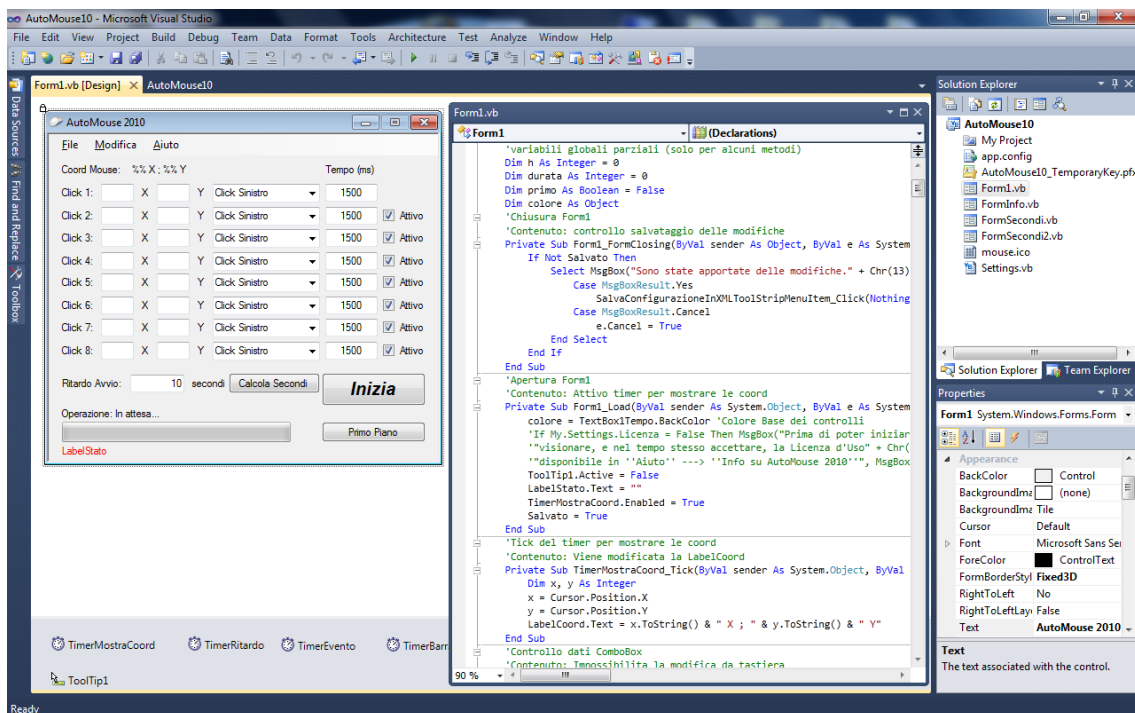
In seguito alla nascita della piattaforma .NET, Visual Studio ha cominciato a supportare diversi linguaggi di programmazione tra i quali ad esempio C#, Visual Basic .NET e C++; non è invece compatibile con Java.

Al suo interno troviamo la tecnologia IntelliSense che permette di correggere eventuali errori sintattici, oltre che alcuni errori logici, senza il bisogno di compilare l'applicazione. Inoltre è dotato di un potente debugger per trovare e correggere gli errori nel codice in runtime e fornisce anche vari strumenti per l'analisi delle prestazioni. Visual Studio si integra nativamente con l'ambiente di sviluppo di gruppo Team Foundation Server, il quale tra le altre cose permette di effettuare operazioni di versioning sul codice.

Differentemente dai classici compilatori, quello del framework .NET converte il codice sorgente del linguaggio di programmazione utilizzato in codice IL intermedio.

La versione 2010 di Visual Studio è stata rilasciata il 12 aprile 2010. Essa presenta alcune importanti novità rispetto alle precedenti, tra cui :

- Il supporto allo sviluppo di applicazioni per il framework .NET 4.0.
- Il nuovo linguaggio di programmazione funzionale F#.
- Il supporto alla programmazione parallela.
- L'integrazione della libreria jQuery.



## 4.3 TORTOISE SVN



TortoiseSVN è un client grafico Subversion ideato per girare come un'estensione di Microsoft Windows ed è gratuito, rilasciato sotto licenza GNU GPL (General Public License).

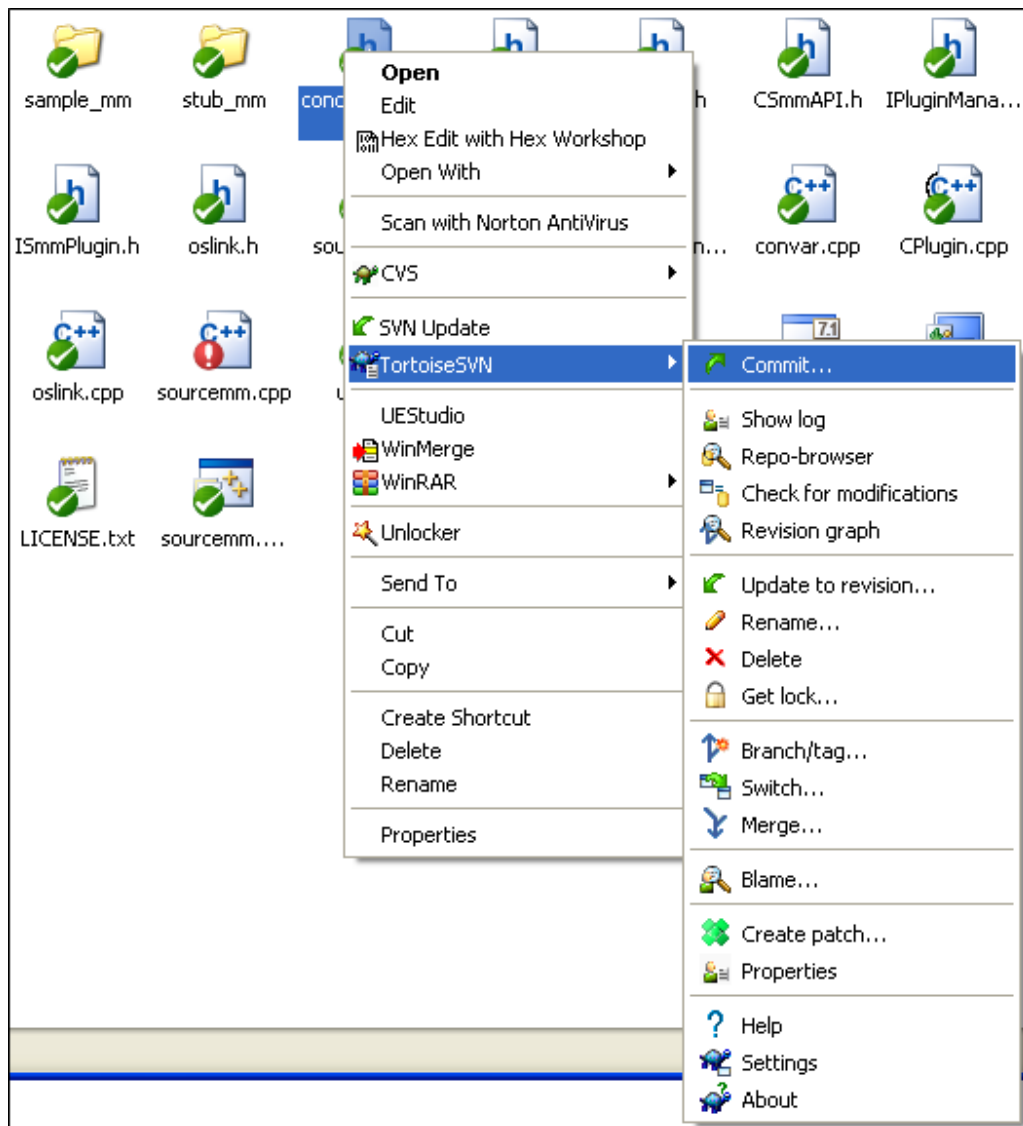
Subversion è un sistema di controllo versione di software ma anche di dati, documenti, foto o file multimediali. E' possibile caricare in remoto i file che fanno parte di un particolare progetto (attraverso un ambiente chiamato repository) in modo che siano a disposizione di tutte le persone che vi stanno lavorando. Solitamente chi lavora ad un progetto effettua in locale le modifiche sui file interessati ed in seguito aggiorna quelli presenti nel repository; il sistema di controllo versione tiene traccia di tutte queste modifiche.

Questo sistema, a differenza del suo predecessore CVS (Concurrent Versioning System) supporta qualsiasi tipo di file.

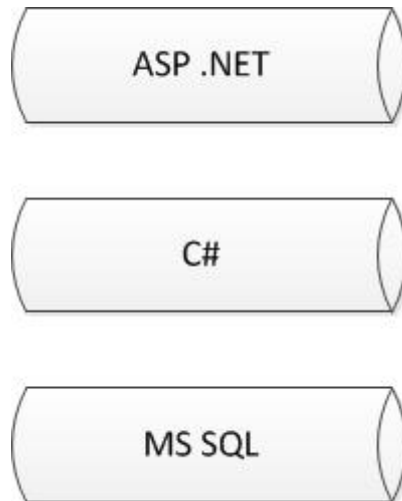
Grazie alle funzionalità di questo software è possibile :

- Tenere traccia di tutte le modifiche effettuate su un determinato file.
- Aprire un file per modificarlo e bloccarne l'accesso a modifiche da parte di altre persone in modo da non creare conflitti di versioni.
- Bloccare eventuali propagazioni di errori e tornare in qualsiasi momento ad una versione precedente.
- Sviluppare versioni parallele del software.
- Eseguire un backup di tutte le versioni del programma in fase di sviluppo in qualsiasi momento.

Come si evince dall'immagine TortoiseSVN si integra con la shell di Windows estendendola con vari comandi. Inoltre viene installato un set di icone che permette di vedere quali sono i file presenti anche nel repository ed il loro stato (aggiornato, non aggiornato, in conflitto).



## 5 LA STRUTTURA DI UN'APPLICAZIONE WEB



Tipicamente un'applicazione web si presenta con una struttura a tre livelli.

Nella maggioranza dei casi è infatti possibile identificare un livello di visualizzazione associabile al browser; un secondo livello costituito dal motore applicativo, scritto in linguaggio di programmazione, ed un terzo livello riconducibile al database associato.

In sintesi, il browser invia le proprie richieste al secondo livello cioè il motore applicativo, il quale interpreta e gestisce le interrogazioni al database e genera un output diretto allo stesso browser, che lo modella e lo restituisce all'utente sotto forma di pagina web.

## 5.1 LA VISUALIZZAZIONE TRAMITE BROWSER

Come è stato più volte ripetuto in precedenza, lo strumento tramite il quale l'utente utilizza l'applicazione web è un semplice browser. Di conseguenza è necessario che vengano sviluppate delle pagine web che contengano al loro interno una vera e propria interfaccia di utilizzo. Per quanto riguarda gli elementi di controllo grafici c'è stata nel corso degli anni una decisa evoluzione verso strutture sempre più complesse, per le quali era un tempo impensabile l'elaborazione all'interno di una pagina web. Allo stesso tempo anche i principali browser si sono evoluti in modo da consentire una fluida esecuzione di questi elementi.

Il linguaggio utilizzato in questo caso, facente parte come le altre tecnologie della famiglia Microsoft, è ASP .NET.



Alcuni loghi di noti browser



### 5.1.1 IL LINGUAGGIO ASP .NET



ASP .NET è un insieme di tecnologie di sviluppo software per il web commercializzate da Microsoft. Con l'utilizzo di queste tecnologie gli sviluppatori possono realizzare applicazioni e servizi web.

Nonostante il nome derivi da ASP (Active Server Pages ovvero la vecchia tecnologia Microsoft per lo sviluppo orientato al web) esistono sostanziali differenze tra i due dato che ASP .NET si basa come tutte le applicazioni della famiglia .NET sul CLR. Le applicazioni in ASP .NET sono nettamente più rapide e garantiscono miglior performance di quelle sviluppate con altre tecnologie di scripting dato che l'intero codice del sito web è precompilato in pochi file dll (a volte addirittura in un unico file) gestiti da un server web. La vasta disponibilità di controlli, classi e strumenti di sviluppo consente di ridurre sensibilmente i tempi di sviluppo dei moduli software e mette lo sviluppatore in condizione di riutilizzare facilmente uno stesso codice per applicazioni differenti.

ASP .NET può contare su molti punti di forza rispetto ad ASP :

- Sono state semplificate significativamente le funzionalità di accesso ai dati, ad esempio è molto più immediato generare una pagina contenente dei dati prelevati da un database.
- Il codice precompilato consente alle applicazioni di girare più rapidamente e di rilevare errori di programmazione già in fase di sviluppo, inoltre è possibile distribuire un'applicazione web senza dover allegare il sorgente.

- I controlli definiti dall'utente consentono di implementare facilmente controlli molto comuni come ad esempio i menu.
- E' possibile impersonare qualunque utente Windows ereditandone tutti i privilegi.
- La possibilità di eseguire una copia cache di intere pagine o di alcune parti migliora notevolmente le performance complessive.
- Lo stato della sessione può essere salvato in modo che i dati della sessione non vengano persi quando l'IIS viene resettato.

ASP .NET ha come obiettivo la semplificazione del passaggio da parte degli sviluppatori dalle applicazioni Windows a quelle web. Infatti li mette in grado di creare pagine composte da tanti controlli widget, in tutto e per tutto analoghi a quelli utilizzati dall'interfaccia utente Microsoft. Un controllo web ha un comportamento analogo a quello della sua controparte Windows : il codice associato può assegnare ad esso delle determinate proprietà ed eseguire azioni in seguito ad eventi ben precisi (come per esempio la pressione di un pulsante). L'unica vera differenza sta nel fatto che mentre i controlli Windows disegnano se stessi sullo schermo, quelli web producono blocchi di codice HTML che vanno ad inserirsi automaticamente all'interno della pagina che deve essere visualizzata dal browser dell'utente.

ASP .NET è progettato in modo da incoraggiare lo sviluppatore ad utilizzare in modo sistematico il paradigma dell'interfaccia grafica (GUI ovvero Graphical User Interface) insieme alla cosiddetta programmazione ad eventi cioè lo stile di programmazione secondo il quale i vari blocchi di codice vengono eseguiti in seguito a determinati eventi o azioni su controlli dotati di rappresentazione grafica.

Le classi del framework .NET possono combinarsi ed interagire con tecnologie esistenti come ad esempio Javascript, in modo tale da attribuire un carattere di persistenza ad oggetti software anche nell'ambito web, che è intrinsecamente privo di stato.

## **5.2 IL MOTORE APPLICATIVO**

Il secondo livello è un core applicativo costituito da codice in un linguaggio di sviluppo dinamico lato server.

Gli sviluppatori possono scrivere codice utilizzando uno qualsiasi dei linguaggi di alto livello supportati dal framework .NET come ad esempio Visual Basic .NET, C#, J# oppure linguaggi open source come Perl e Python, questo è possibile poiché il CLR è standardizzato presso ECMA (European Computer Manufacturers Association).

Naturalmente, il linguaggio di programmazione utilizzato all'interno di questo progetto è il C#, che in un certo senso è il linguaggio che meglio di altri descrive le linee guida sulle quali si basa ogni programma .NET, infatti è stato creato da Microsoft specificatamente per la programmazione nel framework .NET.

### 5.2.1 IL LINGUAGGIO C#

Il C# è un linguaggio di programmazione object-oriented sviluppato da Microsoft all'interno dell'iniziativa .NET.

I suoi tipi di dati primitivi hanno una corrispondenza univoca con i tipi .NET e molte delle sue astrazioni come classi, interfacce ed eccezioni sono particolarmente adatte a gestire il framework .NET. Potremmo definire come linguaggio come interpretato e compilato allo stesso tempo : infatti, data la sua stretta integrazione con il framework, i codici sorgenti sono normalmente compilati secondo i criteri JIT e la trasformazione in codice macchina (eseguito dalla CPU) avviene solamente su richiesta al caricamento e all'esecuzione del programma.

Confrontandolo con il C o con il C++ è possibile notare una serie di modifiche atte ad evitare errori tipici ed ambiguità tipiche della programmazione :

- I puntatori possono essere utilizzati solo in particolari blocchi di codice.
- In molte operazioni aritmetiche vengono controllati gli eventuali overflow.
- E' possibile ereditare da una sola classe ma si possono implementare un numero indefinito di interfacce.
- Le sole conversioni implicite consentite sono quelle sicure ovvero che non espongono al rischio di perdita di dati causata dalla diversità di tipologia.

Il seguente esempio stampa il testo "Hello World" su console :

```
using System;

class HelloWorldApp
{
    public static void Main()
    {
        Console.WriteLine("Hello World");
        Console.ReadLine();
    }
}
```

Il seguente esempio invece definisce una classe di nome Persona che espone una proprietà Nome il cui valore viene memorizzato nella variabile mNome :

```
using System;

namespace Program
{
    Class Persona
    {
        Public string Nome
        {
            get { return mNome; }
            set { mNome = value; }
        }

        Private string nNome;
    }
}
```

### 5.3 IL DATABASE

Un DBMS (Database Management System) è un sistema software progettato per consentire la creazione e la modifica efficiente di database, ovvero di collezioni di dati strutturati, spesso da parte di più utenti.

In passato i DBMS erano diffusi solo all'interno delle grandi aziende (dato anche il notevole costo economico dell'hardware necessario a supportarli), oggi invece il loro utilizzo si è diffuso praticamente in qualsiasi contesto informatico.

L'insieme di programmi software che costituisce un DBMS può essere assai complesso dato che viene controllata l'organizzazione, la memorizzazione, la sicurezza e l'integrità dei dati al suo interno. Inoltre vengono gestite le richieste di dati da parte di qualsiasi programma applicativo e viene istruito il sistema operativo per il trasferimento dei dati appropriati.

Il sistema di sicurezza dei dati impedisce agli utenti non autorizzati di visualizzare o modificare il database. L'accesso all'intero database oppure solamente ad un suo sottoinsieme (in questo caso si parla di subschema) è permesso mediante l'utilizzo di password.

Il DBMS può mantenere l'integrità del database non consentendo a più utenti di modificare il medesimo record contemporaneamente (blocco dei record). Può altresì impedire l'immissione di due record duplicati. L'insieme di regole che determinano l'integrità e la consistenza di una base di dati prendono il nome di vincoli di integrità referenziale.

I linguaggi di interrogazione del database permettono agli utenti di interrogare in maniera interattiva il database e di analizzarne i dati tramite delle interrogazioni chiamate query.

I tre tipi di organizzazione più comuni sono il modello gerarchico, il modello reticolare ed il modello relazionale che è quello oggi dominante, normalmente utilizzato con il linguaggio di interrogazione SQL.

Questo modello assume fundamentalmente che tutti i dati siano rappresentati come relazioni e manipolati con gli operatori dell'algebra relazionale. Con questo modello tutte le informazioni sono rappresentate da valori inseriti in relazioni (tabelle) dunque un database relazionale è un insieme di relazioni contenente valori ed il risultato di qualunque interrogazione dei dati può essere anch'esso rappresentato da relazioni.

I database server sono computer ottimizzati al fine di ospitare i programmi che costituiscono il database reale e sui quali gira solo il DBMS, oltre al software ad esso correlato. Solitamente si tratta di macchine multiprocessore e dotate di dischi fissi configurati in modalità RAID per consentire una memorizzazione stabile ed affidabile dei dati in modo da garantire la continuità dei servizi anche in caso di un guasto ad un componente (in questo caso si parla di "fault tolerant"). In ambienti dove vengono eseguite transazioni con moli di dati particolarmente elevate si utilizzano anche specifici componenti hardware che hanno la funzioni di acceleratori di database i quali sono collegati ai server tramite canali preferenziali ovviamente ad alta velocità.

Recentemente assistiamo sempre più di frequente all'integrazione tra basi di dati e Internet : infatti una vasta classe di applicazioni della rete fa uso di informazioni memorizzate su basi di dati.

In un'applicazione web è necessario avere un sistema di questo tipo in cui memorizzare i dati, in modo che possano venire facilmente richiamati in qualsiasi momento da un elemento caricato dal browser o da codice. Nel nostro caso, dato anche il largo utilizzo di tecnologie Microsoft all'interno del progetto, si è optato per l'utilizzo di MS SQL.

### 5.3.1 IL DBMS MS SQL



Microsoft SQL Server è un DBMS (Database Management System) prodotto da Microsoft. Nelle sue prime edizioni veniva utilizzato solamente per basi di dati medio-piccole ma a partire dalla versione 2000 viene utilizzato anche per gestire basi di dati di notevoli dimensioni.

Le versioni per Windows sono :

<b>ANNO</b>	<b>VERSIONE</b>
1993	SQL Server 4.2
1995	SQL Server 6.0
1996	SQL Server 6.5
1998	SQL Server 7.0
1999	SQL Server 7.0 OLAP
2000	SQL Server 2000 32 bit
2003	SQL Server 2000 64 bit
2005	SQL Server 2005 (32 bit e 64 bit)
2008	SQL Server 2008 (32 bit e 64 bit)
2010	SQL Server 2008 R2 (32 bit e 64 bit)
2012	SQL Server 2012 (32 bit e 64 bit)

Per quanto riguarda all'applicazione web in esame è stata utilizzata la versione 2008 R2 64 bit.



## 6 L'APPLICAZIONE

### TECH GESTIONE STUDIO

Come accennato in precedenza, l'azienda ospitante aveva già sviluppato un software molto semplice orientato al CRM. Su richiesta di uno studio commercialistico ha poi deciso di realizzare un'applicazione più potente e completa, utilizzando le tecnologie di cui si è parlato fino ad ora. Il nome scelto per questa applicazione è Tech Gestione Studio.

In seguito ad una raccolta dei requisiti che è poi proseguita durante la realizzazione del software, è iniziato lo sviluppo vero e proprio.

Gli strumenti offerti dall'applicazione sono raggruppati in quattro menu principali :

- Anagrafiche : la gestione dei clienti e dei contatti.
- Note : la gestione delle note di testo.
- Gestione : la gestione delle pratiche, degli adempimenti e delle prestazioni.
- Amministrazione : la gestione degli utenti che utilizzano l'applicazione.

HOME

ANAGRAFICHE

NOTE

GESTIONE

AMMINISTRAZIONE

## **6.1 GLI OBIETTIVI**

Con Tech Gestione Studio si vuole centralizzare e semplificare la gestione del cliente e di tutte le pratiche ad esso collegate.

Ogni dipendente dell'azienda deve poter accedere con le proprie credenziali e svolgere in autonomia il proprio lavoro, mettendo poi a disposizione degli altri dipendenti il risultato del lavoro svolto.

Si vuole che l'azienda che utilizza questa applicazione possa :

- Avere a disposizione un'unica soluzione integrata per tutte le attività, senza ricorrere ad altri software esterni con conseguenti problemi di compatibilità e funzionamento.
- Tenere sempre a portata di mano tutte le informazioni, i recapiti e le pratiche riguardanti uno specifico cliente dando così una risposta immediata ad ogni richiesta con un risparmio di tempo anche nelle piccole operazioni quotidiane.
- Pianificare con chiarezza il lavoro da dividere tra i vari operatori aumentando così la produttività individuale ed evitando equivoci sull'assegnazione delle prestazioni da effettuare.
- Controllare in qualsiasi momento l'avanzamento delle prestazioni per valutare se i tempi delle pratiche rispettano le previsioni fatte inizialmente o se vi sono dei ritardi, in modo da poter prendere adeguati provvedimenti.
- Riassumere alla fine di un lavoro quante ore sono state spese per una specifica attività, in modo da poterne quantificare il costo in termini economici e di personale e calcolare l'eventuale margine di guadagno.

## 6.2 IL LOGIN



Account Information

Username:  
[ ]

Password:  
[ ]

Keep me logged in

Log In

E' necessario che i dipendenti dell'azienda abbiano la possibilità di effettuare il login con le proprie credenziali, in modo che l'applicazione assegni ad ogni utente che effettua l'accesso i relativi permessi (amministratore, utente avanzato, utente semplice, ecc...). Inoltre è importante la rintracciabilità del nominativo di chi agisce all'interno dell'applicazione, in modo da capire sempre con facilità chi fa una determinata operazione.

Tramite questo semplice codice, presente all'interno di ogni pagina ASP .NET, il sistema controlla ad ogni caricamento di una nuova pagina che l'utente che la sta visualizzando abbia effettuato correttamente il login all'interno dell'applicazione, altrimenti lo reindirizza automaticamente alla pagina di accesso.

```
if (!Request.IsAuthenticated)
    { Response.Redirect("Login.aspx"); }
```

## 6.3 LA PAGINA INIZIALE

La pagina iniziale è ovviamente il primo elemento del menu dell'applicazione e viene caricata subito dopo aver effettuato il login. Essa visualizza un riassunto della situazione attuale.

L'idea di fondo è che un utente, una volta effettuato il login, si trovi davanti le ultime note che sono state create o modificate, in modo da poterle visualizzare immediatamente.

Le note all'interno dell'applicazione sono concepite come istruzioni per gli utenti sul lavoro da svolgere. Visualizzando quelle in rilievo nella pagina principale ogni utente può visualizzare subito ciò che deve fare.

Per questo motivo è presente un controllo ASP .NET che elenca le note create più recentemente. Esso visualizza alcune informazioni come ad esempio la data, la data di avviso della nota, il mittente, il destinatario e l'oggetto, inoltre permette cliccando su quest'ultimo di visualizzare il contenuto della nota, che potrà eventualmente essere modificato. Un'altra funzione fruibile immediatamente da questo controllo è la possibilità di segnare una nota come già letta.

Note recenti					
OGGETTO	DATA	AVVISO	MITTENTE	DESTINATARI	LETTA
nota con lookup validato	21/03/2011	21/03/2011	[REDACTED]	...	<input type="checkbox"/>
tgi	22/03/2011	22/03/2011	[REDACTED]	[REDACTED]	<input type="checkbox"/>
consegnare docs	22/03/2011	25/03/2011	[REDACTED]	[REDACTED]	<input type="checkbox"/>
fare mansione 2	23/03/2011	24/03/2011	[REDACTED]	[REDACTED]	<input type="checkbox"/>
test	15/05/2011	30/04/2011	[REDACTED]	[REDACTED]	<input type="checkbox"/>

Subito dopo è presente un altro controllo ASP .NET del tutto analogo al precedente, il quale però elenca tutte le note non ancora chiuse, ovvero note che contengono istruzioni per operazioni che devono ancora essere svolte. Anch'esso presenta le colonne oggetto ("cliccabile"), data, avviso, mittente e destinatario.

Note non chiuse				
OGGETTO	DATA	AVVISO	MITTENTE	DESTINATARI
<a href="#">primo invio</a>	20/03/2011			...
<a href="#">risposta manuale 2</a>	07/03/2011	07/03/2011		
<a href="#">risposta manuale 1</a>	07/03/2011	07/03/2011		
<a href="#">testRitorno</a>	01/03/2011	01/03/2011		
<a href="#">prima nota con inserimento utenti</a>	27/02/2011	27/02/2011		
<a href="#">test creazione campi</a>	27/02/2011	27/02/2011		
<a href="#">Oggetto 14</a>	14/01/2011	14/02/2011		
<a href="#">Oggetto 13</a>	13/01/2011	13/02/2011		
<a href="#">Oggetto 11</a>	11/01/2011	11/02/2011		
<a href="#">Oggetto 10</a>	10/01/2011	10/02/2011		
<a href="#">Oggetto 9</a>	09/01/2011	09/02/2011		
<a href="#">Oggetto 8</a>	08/01/2011	08/02/2011		
<a href="#">Oggetto 7</a>	07/01/2011	07/02/2011		
<a href="#">Oggetto 6</a>	06/01/2011	06/02/2011		
<a href="#">Oggetto 5</a>	05/01/2011	05/02/2011		
<a href="#">Oggetto 4</a>	04/01/2011	04/02/2011		
<a href="#">Oggetto 3</a>	03/01/2011	03/02/2011		
<a href="#">Oggetto 2</a>	02/01/2011	02/02/2011		
<a href="#">Oggetto 1</a>	01/01/2011	01/02/2011		

Segue poi un piccolo calendario con il quale interagire per poter visualizzare una suddivisione temporale per giorno delle note. E' infatti sufficiente cliccare sul giorno interessato per visualizzare tutte le note presenti con quella data.

Attività e note a calendario						
≤		maggio 2011			≥	
lun	mar	mer	gio	ven	sab	dom
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

A fine pagina è possibile inserire una nuova prestazione di lavoro, inserendo tutti i dati relativi. La filosofia di utilizzo è che una volta svolto un incarico, l'utente inserisca una nuova prestazione e chiuda o modifichi la nota a riguardo. Tutto questo è possibile direttamente dalla pagina principale. Si è deciso di dare molta rilevanza a questo controllo perché l'inserimento di una nuova prestazione è una delle operazioni più frequenti all'interno dell'applicazione.

Una prestazione è una semplice operazione lavorativa svolta da un dipendente dell'azienda.

E' obbligatoria l'inserimento di una descrizione, del lavoratore che ha effettuato la prestazione e della data, questi ultimi due dati di default sono l'utente loggato al momento e la data corrente. Qualsiasi contenuto può comunque essere modificato in seguito.

The screenshot shows a web form titled "Inserimento rapido prestazioni". At the top, there are three tabs: "senza template2" (selected) and "123". The form fields are as follows:

- Descrizione:** An empty text input field.
- Lavoratore:** A text input field containing "filippo" with a search icon and a red 'X' icon.
- Data:** A date picker showing "31/05/2011".
- Durata:** A numeric input field containing "0".
- Tipo:** A dropdown menu with "Seleziona" selected.
- Ora Inizio:** A time input field with a play button icon.
- Ora Fine:** A time input field with a stop button icon.
- Importo Spese:** A numeric input field containing "0".
- Dettaglio Spese:** A large empty text area with a scrollbar.
- Chilometri trasferita:** A numeric input field containing "0".
- Minuti trasferita:** A numeric input field containing "0".
- In sede:** A checkbox that is currently unchecked.

## 6.4 LE ANAGRAFICHE

La gestione delle anagrafiche è una delle parti centrali dell'applicazione ed è la seconda voce del menu della stessa. In questa sezione il concetto fondamentale è la distinzione tra "clienti" e "contatti".

Pensando in riferimento ad una situazione reale, un'azienda è rappresentata dall'idea di cliente mentre i suoi recapiti sono rappresentati dai contatti. Più precisamente il cliente rappresenta l'azienda nel suo complesso, mentre le persone che vi lavorano ad esempio segretarie, dipendenti o altro vengono rappresentate dai contatti.

Quindi all'interno di un cliente sarà possibile memorizzare informazioni generiche quali la ragione sociale o la partita IVA.

All'interno di un contatto sarà invece possibile memorizzare dati più specifici come nome, cognome, ruolo all'interno dell'azienda, ecc.

Ogni contatto deve necessariamente fare riferimento ad un cliente, infatti contiene al suo interno un campo cliente che non può rimanere vuoto. Al contrario un cliente non deve avere per forza dei contatti ad esso associato. Questo comportamento è abbastanza ovvio dato che una persona (ovvero un contatto) deve per forza lavorare all'interno di un'azienda con partita IVA e relativa sede.

Sia i clienti che i contatti possiedono una proprietà stato che può essere attivo o non attivo. A volte è infatti di utilità mantenere le informazioni di clienti o contatti del passato che attualmente non esistono più o con i quali non si hanno più rapporti.

I campi all'interno delle anagrafiche sono personalizzabili a seconda delle esigenze del cliente. E' possibile anche aggiungerne di nuovi, basta creare un attributo corrispondente all'interno del database.

## 6.4.1 I CLIENTI

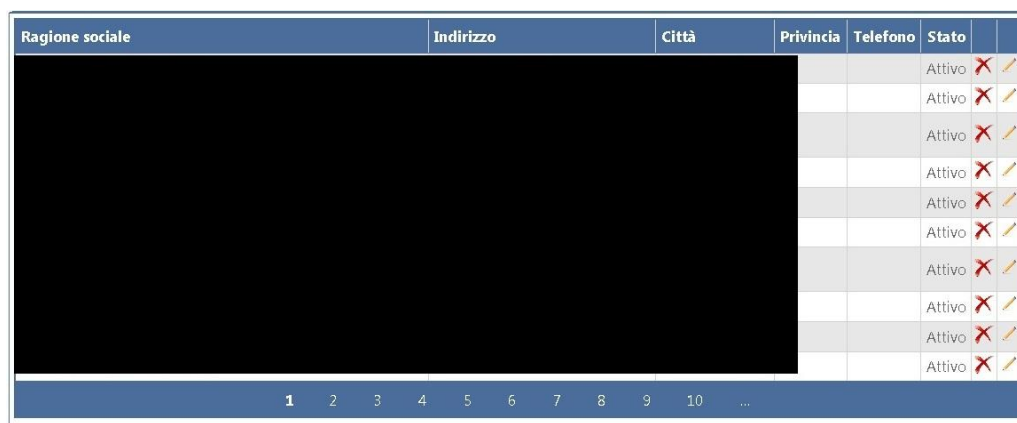
Clienti è la prima voce del sottomenu di anagrafiche.

La pagina presenta inizialmente un controllo ASP .NET per la ricerca dei clienti attraverso campi come la ragione sociale, la città, la provincia, la regione, il numero di telefono oppure lo stato. Dal controllo è anche possibile creare un nuovo cliente.



The screenshot shows a search form titled "Ricerca Clienti" with a blue header. It contains six input fields arranged in two columns. The left column has "Ragione sociale:" (text input), "Regione:" (dropdown menu with "Seleziona" selected), and "Telefono:" (text input). The right column has "Città:" (text input), "Provincia:" (dropdown menu with "Seleziona" selected), and "Stato:" (dropdown menu with "Seleziona" selected). A search icon is visible in the bottom right corner of the form area.

I risultati di questa ricerca verranno visualizzati in un elenco diviso per pagine dove compariranno la ragione sociale, l'indirizzo, la città, la provincia, il numero di telefono e lo stato dei clienti trovati. Sarà inoltre possibile modificarli oppure eliminarli direttamente da tale elenco.



The screenshot shows a table with a blue header and a dark grey body. The header row contains the following columns: "Ragione sociale", "Indirizzo", "Città", "Privincia", "Telefono", "Stato", and a final column with icons. The body of the table is mostly obscured by a black rectangle, but the "Stato" column shows "Attivo" for each row, followed by a red 'X' icon and a pencil icon. At the bottom of the table, there is a pagination bar with numbers 1 through 10 and an ellipsis.

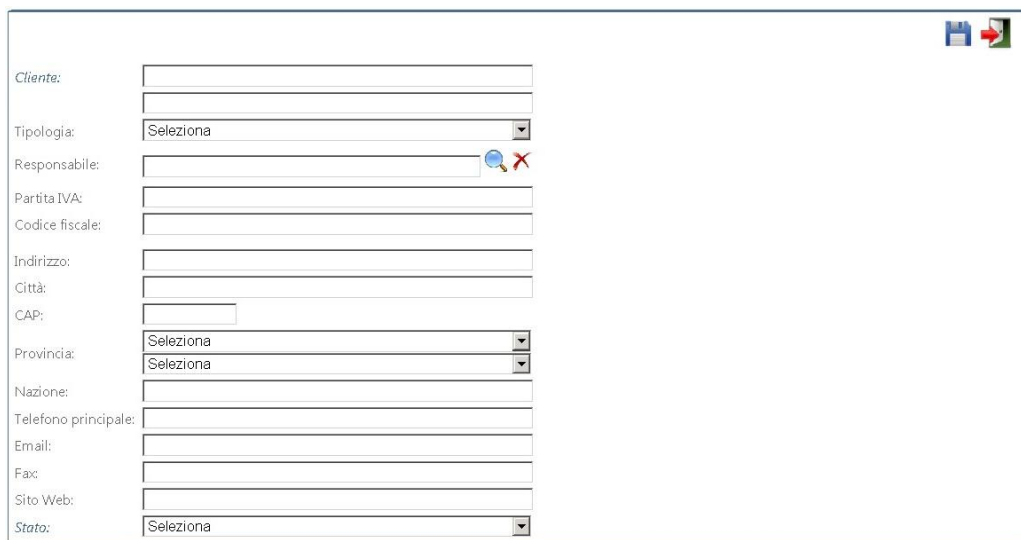
Ragione sociale	Indirizzo	Città	Privincia	Telefono	Stato	
					Attivo	X ✎
					Attivo	X ✎
					Attivo	X ✎
					Attivo	X ✎
					Attivo	X ✎
					Attivo	X ✎
					Attivo	X ✎
					Attivo	X ✎
					Attivo	X ✎
					Attivo	X ✎



La schermata di creazione o di modifica di un cliente presenta i seguenti campi : la ragione sociale, la tipologia, il responsabile, la partita IVA, il codice fiscale, l'indirizzo, la città, il CAP, la provincia, la nazione, il numero di telefono principale, l'indirizzo email, il numero di fax, l'indirizzo del sito web e lo stato.

La ragione sociale e lo stato devono obbligatoriamente essere compilati mentre gli altri campi possono rimanere vuoti.

La tipologia, come la provincia (variabile a seconda della nazione) e lo stato vengono scelti da un elenco che può venire personalizzato a seconda delle esigenze del richiedente, anche in corso di utilizzo dell'applicazione.



The screenshot shows a web form for creating or editing a client. The form is organized into a list of labels on the left and corresponding input fields on the right. The fields include:

- Cliente:** Two empty text input fields.
- Tipologia:** A dropdown menu with the text "Seleziona".
- Responsabile:** A text input field with a search icon (magnifying glass) and a red 'X' icon to its right.
- Partita IVA:** A text input field.
- Codice fiscale:** A text input field.
- Indirizzo:** A text input field.
- Città:** A text input field.
- CAP:** A text input field.
- Provincia:** A dropdown menu with the text "Seleziona".
- Nazione:** A dropdown menu with the text "Seleziona".
- Telefono principale:** A text input field.
- Email:** A text input field.
- Fax:** A text input field.
- Sito Web:** A text input field.
- Stato:** A dropdown menu with the text "Seleziona".

In the top right corner of the form, there are two small icons: a blue floppy disk icon and a green arrow pointing right icon.

All'interno della schermata di modifica vengono anche visualizzati gli eventuali contatti associati, con la possibilità di crearne di nuovi. Questo controllo evita di dover passare per la ricerca contatti, risparmiando una notevole quantità di tempo.



Analogamente, vengono visualizzate le note associate, se esistenti, ed è possibile crearne di nuove. In questo modo è possibile tenere sotto controllo la situazione attuale del cliente, visualizzando tutte le note ad esso relative senza dover passare per forza per la ricerca note.



## 6.4.2 I CONTATTI

Contatti è la seconda voce del sottomenu di anagrafiche.

La pagina si presenta con un controllo ASP .NET per la ricerca dei contatti. La ricerca può svolgersi attraverso campi come il nome, il cognome, il cliente associato (che viene selezionato da un elenco che appare all'interno di un pop-up) oppure lo stato. Dal controllo è anche possibile creare un nuovo contatto.



I risultati della ricerca compaiono poi in un elenco diviso per pagine che mostra la ragione sociale del cliente associato, il nome, il cognome, il numero di telefono, l'indirizzo email e lo stato. Dallo stesso elenco è possibile modificare o eliminare un contatto.

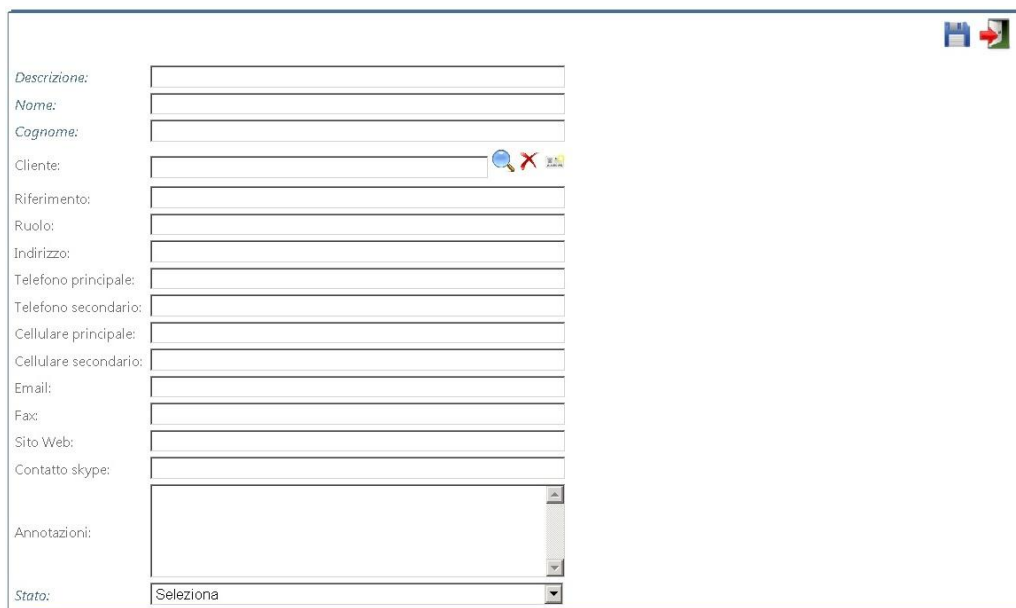
Ragione sociale	Nome	Cognome	Telefono	Cellulare	Email	Stato		
						Attivo	X	
						Attivo	X	
						Attivo	X	
						Attivo	X	
						Attivo	X	
						Attivo	X	
						Attivo	X	
						Attivo	X	
						Attivo	X	
						Attivo	X	

1 2 3 4 5 6 7 8 9 10 ...

La schermata di modifica, del tutto analoga a quella del cliente, presenta i seguenti campi : la descrizione, il nome, il cognome, il cliente associato, il riferimento, il ruolo, l'indirizzo, il numero di telefono principale, il numero di telefono secondario, il numero di cellulare principale, il numero di cellulare secondario, l'indirizzo email, il numero di fax, l'indirizzo del sito web, il contatto Skype, le eventuali annotazioni e lo stato.

La descrizione, il nome, il cognome e lo stato, al contrario degli altri campi, sono obbligatori.

Il cliente associato viene scelto tramite un elenco dei clienti già esistenti che si apre tramite un pop-up.



The screenshot shows a web form for editing a user profile. The form is organized into a list of labels on the left and corresponding input fields on the right. The labels are: *Descrizione:*, *Nome:*, *Cognome:*, *Cliente:*, *Riferimento:*, *Ruolo:*, *Indirizzo:*, *Telefono principale:*, *Telefono secondario:*, *Cellulare principale:*, *Cellulare secondario:*, *Email:*, *Fax:*, *Sito Web:*, *Contatto skype:*, *Annotazioni:*, and *Stato:*. The *Cliente:* field includes a search icon (magnifying glass) and a red 'X' icon. The *Annotazioni:* field is a large text area with a vertical scrollbar. The *Stato:* field is a dropdown menu with the text 'Seleziona' visible. In the top right corner of the form, there are two small icons: a blue folder icon and a red arrow icon.

Come nel caso dei clienti, all'interno della schermata di modifica è possibile visualizzare le eventuali note associate oppure crearne di nuove.



## 6.5 LE NOTE

La gestione delle note è il terzo elemento del menu e non presenta alcun sottomenu al suo interno.

Una nota deve sempre avere un oggetto, un contenuto di testo, un destinatario (che può essere indifferentemente un cliente o un contatto), oltre alla data e al mittente che si memorizzano automaticamente. Inoltre è possibile fissare una data affinché compaia un promemoria al destinatario della nota.

Una nota può avere lo stato di aperta o chiusa, a seconda del fatto che le istruzioni al suo interno siano state eseguite oppure no.

Nella gestione delle note compare un controllo ASP .NET che permette di ricercare le note presenti nel database tramite l'oggetto, il testo oppure l'intervallo temporale della creazione o dell'avviso della nota. Inoltre è possibile creare una nuova nota seduta stante grazie all'apposito pulsante.



Analogamente a quanto avviene nella gestione delle anagrafiche, quando viene effettuata una ricerca i risultati vengono visualizzati in un elenco diviso in pagine dove compare l'oggetto, il testo, il mittente e l'icona di stato.

Dall'elenco è inoltre possibile visualizzare, modificare o cambiare lo stato di una nota.

Oggetto	Testo	Letta	Mittente			
<a href="#">fare mansione 2</a>	<a href="#">ricordati di finire i tuoi compiti per domani</a>	<input type="checkbox"/>				
<a href="#">risposta manuale 1</a>	<a href="#">risposta manuale 1</a>	<input type="checkbox"/>				
<a href="#">test creazione campi</a>	<a href="#">test creazione campi audit</a>	<input type="checkbox"/>				
<a href="#">primo invio</a>	<a href="#">test</a>	<input type="checkbox"/>				
<a href="#">Oggetto 13</a>	<a href="#">Descrizione 13</a>	<input type="checkbox"/>				
<a href="#">nota con lookup validato</a>	<a href="#">nota con lookup validato</a>	<input type="checkbox"/>				
<a href="#">Oggetto 11</a>	<a href="#">Descrizione 11</a>	<input type="checkbox"/>				

Dalla schermata di modifica di una nota è possibile modificare l'oggetto, la descrizione, la data, la data di promemoria, i destinatari (i quali vengono selezionati da un elenco degli utenti registrati al sistema che si apre tramite un pop-up), lo stato, lo stato di lettura e l'area di riferimento dell'azienda.

L'oggetto, la descrizione, la data, la data di promemoria e lo stato sono campi obbligatori.

La data e la data di promemoria di default sono quelle del giorno in cui viene creata la nota mentre il mittente di default è l'utente loggato al momento della creazione.

Oggetto:

Descrizione:

Data:

Avviso:

Mittente:

Destinatari:

Stato:

Letta:

Area:

## 6.6 LA GESTIONE

La gestione del lavoro è il quarto elemento del menu dell'applicazione e presenta un sottomenu con tre voci : pratiche, adempimenti e prestazioni.

Ogni pratica deve essere associata ad un cliente, essa è suddivisa in adempimenti che sono a loro volta suddivisi in singole prestazioni. Questa suddivisione aumenta il livello di precisione della descrizione di un lavoro. Come per altri concetti dell'applicazione, una pratica possiede una proprietà detta stato che può essere attivo o non attivo. Questo perché in un'azienda può capitare che una pratica venga ad esempio sospesa per un certo periodo. Un'altra informazione importante della pratica è la data di scadenza, che indica all'utente il giorno entro il quale la pratica dev'essere conclusa.

Ogni adempimento, com'è logico che sia, dev'essere associato ad una pratica e quindi di conseguenza ad un cliente. Inoltre deve avere un assegnatario ed eventualmente può esserci anche un supervisore. L'utente che crea l'adempimento è di default l'assegnatario, ma può essere modificato in seguito.

Proseguendo nella logica di questo incapsulamento, ogni prestazione deve essere associata ad un adempimento (di conseguenza ad una pratica e ad un cliente). Inoltre l'utente che la effettua è chiamato nell'applicazione lavoratore e dev'essere un utente presente nel sistema (chi crea la prestazione è di default il lavoratore, ma può essere modificato in seguito).




## 6.6.1 LE PRATICHE

Pratiche è la prima voce del sottomenu della gestione.

Nella pagina è presente un controllo ASP .NET per la ricerca della pratiche, attraverso i clienti ad esse associate oppure il loro stato che può essere attivo o non attivo. E' inoltre possibile creare una nuova pratica.



Come avviene analogamente per le anagrafiche e le note, i risultati di questa ricerca vengono visualizzati in un elenco diviso in pagine dove vengono visualizzati la descrizione ed il cliente relativi alla pratica. Dallo stesso elenco è poi possibile passare alla schermata di modifica.

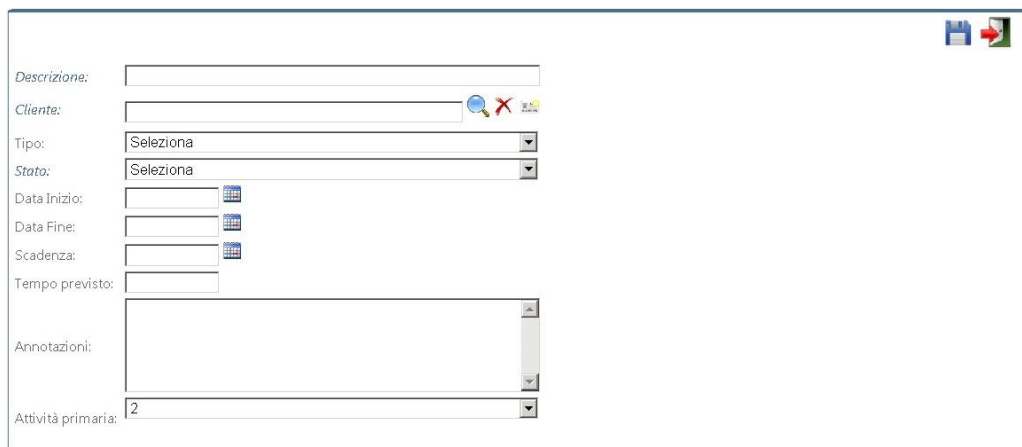
Descrizione	Cliente	Modello	
<a href="#">senza_template2</a>	test		
<a href="#">Boh</a>			
<a href="#">senza_template</a>	test		



Nella schermata di modifica è possibile compilare i seguenti campi : la descrizione, il cliente (il quale viene scelto da un elenco di clienti già esistenti che si apre in un pop-up), il tipo, lo stato, la data di inizio, la data di fine, la scadenza, il tempo previsto, le annotazioni e l'ordine di importanza in confronto alle altre pratiche relative allo stesso cliente.

La descrizione, il cliente associato e lo stato sono campi obbligatori.

Il tipo viene scelto da un elenco che può essere personalizzato a seconda delle esigenze del richiedente, anche durante l'utilizzo dell'applicazione.



The screenshot shows a web-based form for editing a record. The form is organized into several sections:

- Descrizione:** A single-line text input field.
- Cliente:** A text input field with a search icon (magnifying glass), a refresh icon (circular arrow), and a delete icon (X).
- Tipo:** A dropdown menu with the text "Seleziona" and a downward arrow.
- Stato:** A dropdown menu with the text "Seleziona" and a downward arrow.
- Data Inizio:** A text input field with a calendar icon.
- Data Fine:** A text input field with a calendar icon.
- Scadenza:** A text input field with a calendar icon.
- Tempo previsto:** A text input field.
- Annotazioni:** A multi-line text area with a scroll bar.
- Attività primaria:** A dropdown menu with the value "2" and a downward arrow.

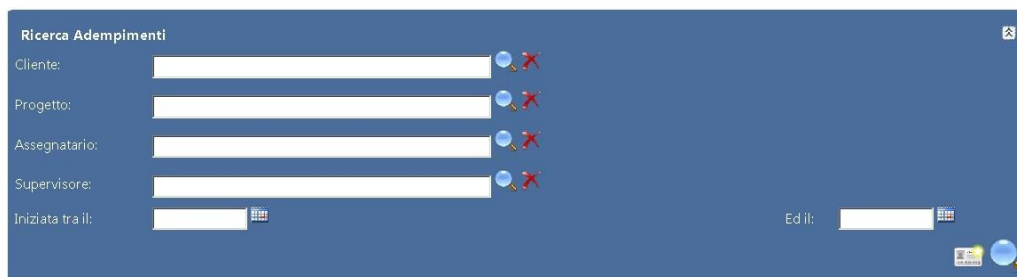
In the top right corner of the form, there are two icons: a blue save icon and a red refresh icon.

## 6.6.2 GLI ADEMPIMENTI

Adempimenti è la seconda voce del sottomenu Gestione.


Dal controllo ASP .NET di ricerca è possibile cercare gli adempimenti per cliente, pratica, assegnatario, supervisore oppure data.

Inoltre dal medesimo controllo è possibile creare un nuovo adempimento.



I risultati così ottenuti vengono divisi in pagine ed è possibile visualizzare immediatamente la data, il titolo, la data di consegna ed il progresso. Il progresso è la percentuale di completamento dell'adempimento e viene modificato a discrezione dell'utente che ci sta lavorando.

Dall'elenco è ovviamente possibile visualizzare per intero un singolo adempimento (cliccando sul titolo) oppure modificarlo.

Data	Titolo	Scadenza	Consegna	Progresso	
<a href="#">30/03/2011</a>	<a href="#">fdsfs</a>	<a href="#">17/03/2011</a>	<a href="#">23/03/2011</a>	53	
<a href="#">23/03/2011</a>	<a href="#">123</a>	<a href="#">23/03/2011</a>	<a href="#">30/03/2011</a>	10	
<a href="#">23/03/2011</a>	<a href="#">TechIrpef</a>	<a href="#">01/05/2011</a>	<a href="#">15/05/2011</a>	15	
<a href="#">23/03/2011</a>	<a href="#">documentazion</a>	<a href="#">27/03/2011</a>	<a href="#">31/03/2011</a>	10	
<a href="#">31/03/2011</a>	<a href="#">fgfd</a>	<a href="#">02/03/2011</a>	<a href="#">17/03/2011</a>	52	
<a href="#">22/03/2011</a>	<a href="#">123</a>	<a href="#">28/03/2011</a>	<a href="#">31/03/2011</a>	30	

Nella schermata di modifica di un adempimento è possibile modificare la priorità, il titolo, la descrizione, la data di consegna, la pratica di appartenenza, il cliente a cui si riferisce, il progresso in percentuale, l'assegnatario ed il supervisore.

Titolo, descrizione, pratica di appartenenza, cliente, progresso ed assegnatario sono campi obbligatori. La pratica, il cliente, l'assegnatario ed il supervisore possono essere scelti tra quelli esistenti da un elenco che si apre tramite un pop-up.

The screenshot shows a web form for editing a task. The form is contained within a rectangular frame with a blue border. In the top right corner of the frame, there are two small icons: a blue floppy disk and a red arrow pointing right. The form fields are as follows:

- Categoria:** A dropdown menu with the value '2' selected.
- Titolo:** A text input field.
- Descrizione attività:** A dropdown menu.
- Data di consegna:** A date input field with a calendar icon.
- Data di chiusura:** A date input field with a calendar icon.
- Pratica:** A text input field with a magnifying glass icon and a red 'X' icon to its right.
- Cliente:** A text input field with a magnifying glass icon and a red 'X' icon to its right.
- Progresso:** A slider control with a numerical input field containing the value '0'.
- Assegnatario:** A text input field containing the name 'filippo', with a magnifying glass icon and a red 'X' icon to its right.
- Supervisore:** A text input field with a magnifying glass icon and a red 'X' icon to its right.

### 6.6.3 LE PRESTAZIONI



Prestazioni è il terzo elemento del sottomenu gestione.

Il controllo ASP .NET per la ricerca permette di trovare prestazioni tramite il cliente, il lavoratore oppure la data.

Dallo stesso controllo è anche possibile creare una nuova prestazione (tale operazione è possibile anche dalla pagina principale dell'applicazione).



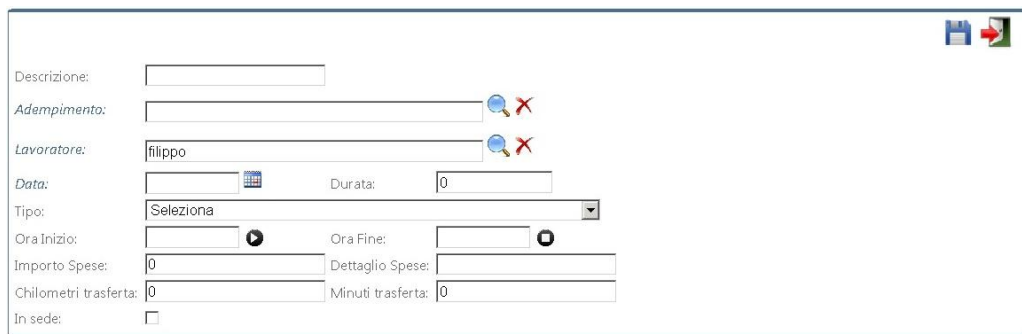
I risultati della ricerca verranno elencati mostrando la descrizione, la data e la durata in minuti delle singole prestazioni, con la possibilità di visualizzarle, modificarle o eliminarle direttamente da tale elenco.

Descrizione	Data	Minuti dedicati		
ghhf	<a href="#">22/03/2011</a>	<a href="#">392</a>		

Nella schermata di modifica della prestazione è possibile modificare la descrizione, l'adempimento a cui si riferisce, il lavoratore che la effettua, la data, la durata, il tipo, l'ora di inizio e di fine, le spese e l'eventuale distanza della trasferta.

Adempimento, lavoratore e data sono campi obbligatori. Adempimento e lavoratore possono essere scelti tra quelli esistenti tramite un elenco che si apre in un pop-up. Due comodi pulsanti possono riempire i campi relativi all'ora con l'orario corrente.

Come già visto in precedenza, il medesimo controllo è presente anche nella pagina principale.



The screenshot shows a web form for editing a performance record. The form contains the following fields and controls:

- Descrizione:** A text input field.
- Adempimento:** A text input field with a search icon and a red 'X' icon.
- Lavoratore:** A text input field containing the name "filippo" with a search icon and a red 'X' icon.
- Data:** A date picker icon followed by a text input field.
- Durata:** A text input field containing the value "0".
- Tipo:** A dropdown menu with the text "Seleziona" and a downward arrow.
- Ora Inizio:** A text input field with a play button icon.
- Ora Fine:** A text input field with a play button icon.
- Importo Spese:** A text input field containing the value "0".
- Dettaglio Spese:** A text input field.
- Chilometri trasferta:** A text input field containing the value "0".
- Minuti trasferta:** A text input field containing the value "0".
- In sede:** A checkbox.

In the top right corner of the form, there are two icons: a blue floppy disk icon and a red arrow icon pointing to the right.

## 6.7 L'AMMINISTRAZIONE

Amministrazione è l'ultimo elemento del menu principale.

Tramite il controllo ASP .NET è possibile ricercare gli utenti per username, indirizzo email, data di creazione o data dell'ultima attività.

Nello stesso controllo è presente anche il comando per creare un nuovo utente.



I risultati della ricerca vengono visualizzati in un elenco diviso per pagine dove è possibile visualizzare lo username, l'indirizzo email, la data di creazione e l'ultimo accesso. Inoltre dall'elenco è possibile visualizzare, modificare o eliminare un singolo utente.

Login	Email	Creato il	Ultimo accesso		
[REDACTED]	[REDACTED]	10/01/2011	10/01/2011	X	✎
filippo	fi@f.net	14/02/2011	14/02/2011	X	✎
[REDACTED]	g@g.ne	22/01/2011	22/01/2011	X	✎
internal	internal@test.it	07/03/2011	07/03/2011	X	✎
[REDACTED]	g@g.com	22/03/2011	22/03/2011	X	✎
r	r	14/02/2011	14/02/2011	X	✎
super	super@test.it	07/03/2011	07/03/2011	X	✎
user	user@test.it	07/03/2011	07/03/2011	X	✎
viewer	viewer@test.it	07/03/2011	07/03/2011	X	✎

La schermata di modifica dell'utente è molto semplice. I campi presenti sono lo username, la password (con un ulteriore campo di conferma) e l'indirizzo email.

E' poi presente un modulo per assegnare all'utente il livello di permessi desiderato. Questi livelli sono ovviamente personalizzabili a seconda delle esigenze del richiedente.

Ovviamente, eccetto diversa richiesta del cliente, solamente l'amministratore può assegnare i permessi, sia in fase di creazione dell'utente che in seguito.



The screenshot displays a user modification interface. On the left, there are four input fields: 'Login:' with the value 'filippo', 'Password:', 'Conferma Password:', and 'Email:' with the value 'fr@f.net'. On the right, there are two panels for role assignment. The 'Ruoli assegnati' panel shows 'ADMIN' with a right-pointing arrow. The 'Ruoli assegnabili' panel lists four roles: 'INTERNAL ADMIN', 'SUPERVISOR', 'USER', and 'VIEWER', each with a left-pointing arrow.

Ruoli assegnati	
ADMIN	→

Ruoli assegnabili	
←	INTERNAL ADMIN
←	SUPERVISOR
←	USER
←	VIEWER

## **7 CONCLUSIONI**

L'azienda si era proposta di realizzare un'applicazione web dotata di funzionalità avanzate per la gestione clienti.

La scelta di appoggiarsi alla piattaforma Microsoft .NET è stata positiva, in quanto, grazie all'esperienza già acquisita in passato dai dipendenti con questa tecnologia, non ci sono stati grossi imprevisti nel cammino di progettazione.

Un confronto continuo con il richiedente ha fatto sì che l'applicazione venisse adattata alle sue esigenze in corso d'opera. L'applicazione può essere considerata una base ben sviluppata, perfettamente adatta anche a sviluppi futuri basati su specifiche richieste di altri clienti.

Uno degli obiettivi principali era senza dubbio la facilità d'uso nonostante la possibilità di mettere a disposizione funzionalità avanzate. Tale obiettivo è stato sicuramente raggiunto grazie ad un'interfaccia intuitiva e alla presenza di numerose scorciatoie che rendono l'utilizzo del software più semplice e rapido per l'utente.

Già adesso l'applicazione può essere accessibile via internet tramite qualunque dispositivo dotato di browser. Un interessante sviluppo futuro potrebbe essere il perfezionamento dell'interfaccia per l'accesso tramite cellulari o altri dispositivi mobili. Si potrebbe anche ipotizzare la realizzazione di un'applicazione vera e propria ottimizzata per il funzionamento con sistemi operativi per smartphone come ad esempio iOS oppure Android.



## **BIBLIOGRAFIA**

- Principi di Ingegneria del software (Roger S. Pressman, V Edizione, McGraw-Hill)
- C# 2008 – Guida per lo sviluppatore (Nagel C. – Hoepli)
- Realizzazione di applicazioni informatiche per l'accesso alle funzionalità di ufficio in mobilità (tesi di laurea triennale di Luca Gasparini, Ingegneria Informatica, Padova 2010)
- Wikipedia – L'enciclopedia libera (<http://it.wikipedia.org>)
- Microsoft .NET (<http://www.microsoft.com/net>)
- MSDN Library (<http://msdn.microsoft.com>)
- Microsoft Visual Studio (<http://www.microsoft.com/visualstudio>)
- TortoiseSVN (<http://tortoisesvn.net>)
- Sito personale di Carlo Becchi (<http://www.deathlord.it>)

## **RINGRAZIAMENTI**

Un grande ringraziamento va al mio relatore : il professor Dalpasso, che mi ha seguito con costanza nella stesura di questa tesi. Ringrazio poi il sig.Girardi, titolare di Techgest, per avermi accolto nella sua azienda durante lo svolgimento del tirocinio ed i colleghi Jessica, Elisa, Nicola, Paolo e Marco che mi hanno costantemente aiutato.

Un particolare ringraziamento va alla mia famiglia : ai miei genitori Lina e Rino e alle mie sorelle Giovanna e Serena. Non posso poi scordare i colleghi dell'università di Treviso, in particolare Alessandro, Alberto, Francesco, Fabio ed Enrico. Un grazie anche a tutti gli amici della mia compagnia, che hanno reso molto piacevoli questi anni di studi.