

# UNIVERSITÁ DEGLI STUDI DI CAMERINO

**Scuola di Scienze e Tecnologie**

*Corso di Laurea in Informatica*



## **Studio ed implementazione di un applicativo software per la gestione personalizzata di dati in ambito medico**

Curriculum Tecnologie Informatiche  
Tesi di Laurea Progettuale

*Studente:*

**Daniele Lionetti**

*Matricola:*

**072451**

*Relatore:*

**Prof. Fausto Marcantoni**

*Correlatore:*

**Prof. Michele Germani**

---

*anno accademico 2013/2014*



## *Ringraziamenti:*

*Il primo ringraziamento va alla mia famiglia:*

*mio padre Nicola, mia madre Luigia, ai miei fratelli Antonio e Raffaele, e mia cognata Elle che nonostante le difficoltà che si sono presentate hanno sempre creduto in me, e mi hanno sempre supportato durante questo viaggio. Grazie alla mia nipotina Isabella, che senza saperlo mi ha dato una gioia indescrivibile.*

*Il secondo ringraziamento va ai coinquilini avuti nel corso degli anni, sono stati per me come una seconda famiglia:*

*Fabrizio, che con me ha avuto una pazienza enorme, Marco e Mattia C., amici di una vita. Davide e Claudio, con cui ho passato esperienze bellissime, infine Michele Chiara e Catia, in questi anni di convivenza mi hanno dato molto più di quanto io possa aver dato loro. Ringrazio gli amici di Camerino con il quale ho potuto confrontarmi senza pregiudizi.*

*Vorrei anche ringraziare tutti gli amici di Falconara, quelli di una vita, per avermi sempre spronato a tenere duro nonostante le avversità che la vita ci chiede di affrontare, e gli amici arrivati dopo, che mi hanno regalato momenti di spensieratezza quando ero in difficoltà.*

*Ringrazio il relatore Prof. Marcantoni e il correlatore prof Germani per l' aiuto dato nella tesi e nello stage*

*Un ultimo sentito ringraziamento va ad Hyperlean e al suo team di sviluppo e in particolare al collega e caro amico Mattia P. per avermi dato la possibilità di mettere in gioco la mia persona e migliorarmi professionalmente e umanamente.*

<u>INTRODUZIONE</u> .....	5
1. <u>OBBIETTIVI DEL PROGETTO</u> .....	6
2. <u>CAPITOLO 1: ELEMENTI PROGETTUALI</u> .....	8
2.1. <u>MICROSOFT ACCESS</u> .....	8
2.2. <u>SQL SERVER 2008</u> .....	13
2.3. <u>VISUAL STUDIO 2008</u> .....	17
2.4. <u>SQL MIGRATION ASSISTANT FOR ACCESS</u> .....	22
2.5. <u>ACTIVE DIRECTORY</u> .....	23
2.6. <u>WINDOWS SERVER 2008</u> .....	25
2.7. <u>INTEROP WORD ACCESS PER VISUAL STUDIO</u> .....	26
2.8. <u>DLL AZIENDALI CUSTOM</u> .....	28
3. <u>CAPITOLO 2: SVILUPPO DEL SOFTWARE E CONFIGURAZIONE</u> .....	30
3.1. <u>BASE DI DATI</u> .....	31
3.2. <u>CONCETTO DI ASTRAZIONE TDA</u> .....	35
3.3. <u>SVILUPPO CODICE E GESTIONE DELLA STRUTTURA</u> .....	37
3.3.1.1. <u>Operazione di Inserimento</u> .....	40
3.3.1.2. <u>Operazione di Modifica</u> .....	41
3.3.1.3. <u>Operazione di Cancellazione</u> .....	43
4. <u>CAPITOLO 3: TESTING</u> .....	48
5. <u>CAPITOLO 4: CONCLUSIONI</u> .....	49
6. <u>APPENDICE</u> .....	50
7. <u>BIBLIOGRAFIA</u> .....	58

# -INTRODUZIONE

## DESCRIZIONE DEL CONTESTO

Il ruolo dei dispositivi medici nell' ambito dell' assistenza sanitaria è diventata, nel corso degli anni, di grande importanza. La varietà e la spinta di questo settore hanno inciso in maniera significativa nell' aumentare la qualità e l' efficienza dell' assistenza sanitaria. Il settore è molto vasto, e parte dalle produzioni di semplici bende fino a complessi macchinari computerizzati.

### *definizione legislativa di medical device*

Per Dispositivo Medico o Medical Device s'intende "Qualunque strumento, apparecchio, impianto, software sostanza o altro prodotto, utilizzato da solo o in combinazione, compreso il software destinato dal fabbricante ad essere impiegato specificamente con finalità diagnostiche o terapeutiche e necessario al corretto funzionamento del dispositivo, destinato dal fabbricante ad essere impiegato sull'uomo a scopo di diagnosi prevenzione, controllo, terapia o attenuazione di una malattia; di diagnosi, controllo, terapia, attenuazione o compensazione di una ferita o di un handicap; di studio, sostituzione o modifica dell'anatomia o di un processo fisiologico; di intervento sul concepimento, il quale prodotto non eserciti l'azione principale, nel o sul corpo umano, cui è destinato, con mezzi farmacologici o immunologici né mediante processo metabolico ma la cui funzione possa essere coadiuvata da tali mezzi;"(D.Lgs 46/1997).

Per quanto riguarda l'aspetto dei software utilizzati nel settore medico, la classificazione come medical device di questi, è regolata dalla direttiva 93/42/CE. Questa direttiva, individua tre tipologie di software:

1. Soggetto a marcatura CE (ES: analisi segnali cardiaci, calcolo stima e modellizzazione di posizionamenti chirurgici)
2. Non soggetto a marcatura CE (ES: software integrati in una pompa di infusione)

Software escluso dalla direttiva CE (ES: anagrafica pazienti, software per formazione di personale medico, software per manutenzione di dispositivi medici o componenti.)

## -OBBIETTIVI

L' obiettivo principale è stato quello di realizzare un software per la gestione dell'anagrafica pazienti e la rielaborazione dei dati clinici a scopo statistico e di ricerca. Quindi in base a questi a prerequisiti si è deciso di sviluppare un software che fa parte della terza tipologia. I prodotti di questa tipologia sono definiti prodotti borderline.

### *prodotti borderline*

I prodotti borderline sono quei prodotti che per loro natura non appartengono con chiarezza ad un determinato settore, per i quali quindi è difficile definire quale sia la normativa di riferimento da applicare.

In alternativa prodotti borderline sono quei prodotti che rientrano nella definizione di dispositivo medico, ma sono esclusi dal campo di applicazione delle pertinenti direttive. Quando un prodotto non rientra nella definizione di dispositivo medico riportata nella direttiva o è escluso dal campo di applicazione della direttiva stessa, si applicano altre direttive o normative di riferimento. Un prodotto borderline può ricadere nell'ambito di applicazione delle direttive sui dispositivi medici di altre direttive. Questi prodotti, di difficile collocazione sono definiti borderline in quanto sono al limite tra la definizione di dispositivo e le definizioni fornite da normative che regolano altri settori. La destinazione d'uso del dispositivo medico che, comunque, deve essere in ogni caso connotabile con una finalità medica, dovrebbe dirimere ogni dubbio di demarcazione con prodotti che non devono vantare tale finalità. Spetta comunque al fabbricante decidere se il proprio prodotto rientra nella definizione di dispositivo medico (Decreto legislativo 46/97 art. 1 comma 2). In caso di dubbio saranno le Autorità competenti a decidere sulla collocazione dei prodotti borderline. A tal fine è stato istituito presso la Commissione Europea un gruppo di lavoro Borderline and Classification medical devices expert\_group.

Definita la parte di classificazione del software, nel secondo obiettivo si è scelto di concentrarsi sulla parte di sviluppo del software. Particolare attenzione è stata prestata alla progettazione della struttura dati.

Nell'implementazione della struttura dati e nelle relazioni si è cercato di rispettare questi requisiti, ciò che caratterizza il Db è :

- La struttura dati

#### la ridondanza minima

i dati non devono essere inutilmente duplicati per problemi di:

- spazio
- gestione
- manutenzione
- affidabilità
- coerenza

#### la permanenza dei dati

la base di dati è protetta contro eventi che possano minacciarne l'esistenza e/o l'integrità.

#### la condivisione dei dati

più utenti devono potere ad un tempo usare la stessa base di dati (supporto unico, aggiornamento unico, coerenza dei dati, affidabilità, ...)

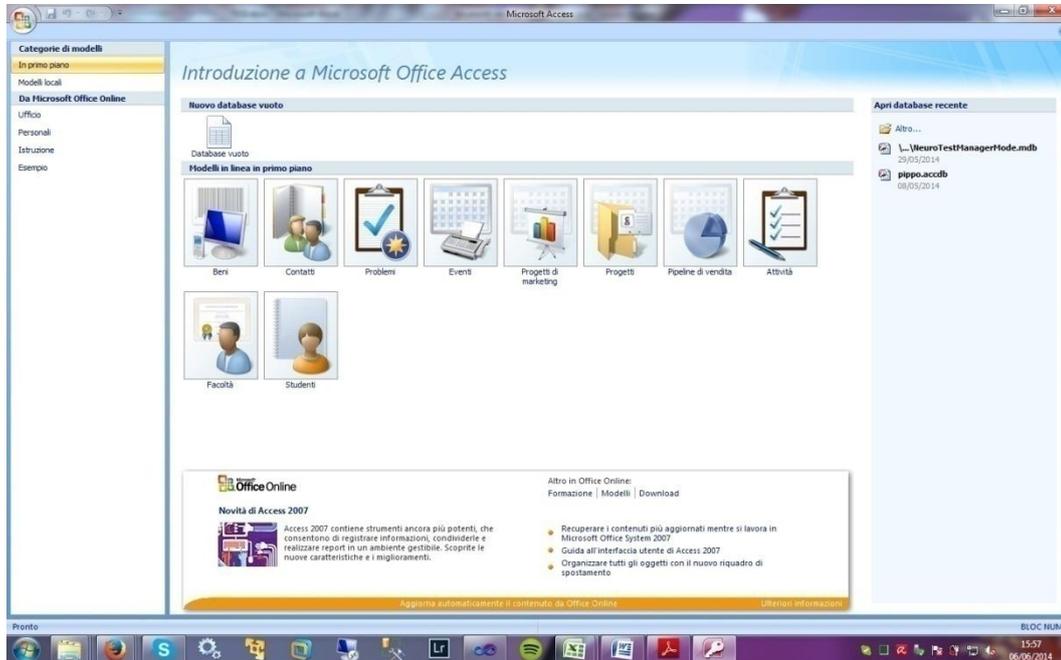
- Le relazioni fra i dati

#### Affidabilità dati

Il database è composto da un numero consistente di tabelle, perché la maggior parte di queste rappresentano le visite e i test che vengono effettuati sui pazienti. E difficile garantire una buona affidabilità e facilità di manutenzione a strutture di dati di questo tipo che necessitano di numerose relazioni. Non è possibile a priori fare a meno di queste relazioni, ma si può rendere più efficiente la struttura portando la logica di queste, dal database al codice, rendendo più flessibile robusta e affidabile la base di dati.

# Capitolo 1: ELEMENTI PROGETTUALI:

## - Descrizione di Microsoft Access



*schermata principale di Microsoft Access*

Microsoft Access è un'interfaccia software atta a gestire le basi di dati relazionali RDBMS (Relational Database Management System) e che integra anche il modulo per lo sviluppo rapido di software applicativi gestionali di piccole e medie dimensioni con metodologia RAD (Rapid Application Development).

## Formato dati

Microsoft Access può usare dati immagazzinare dati in formato access e qualsiasi altra base di dati in formato compatibile ODBC. La struttura segue il modello tabella relazionale, cioè immagazzina i dati da gestire in tabelle composte da un certo numero di record, e i record stessi contengono dati distinti per campi.

Le tabelle sono collegate tra loro tramite la *relazione* che consente di fare associazioni tra i dati collegando le varie tabelle, consentendo di avere una pluralità dei dati anche complessa.

## Memorizzazione Dati

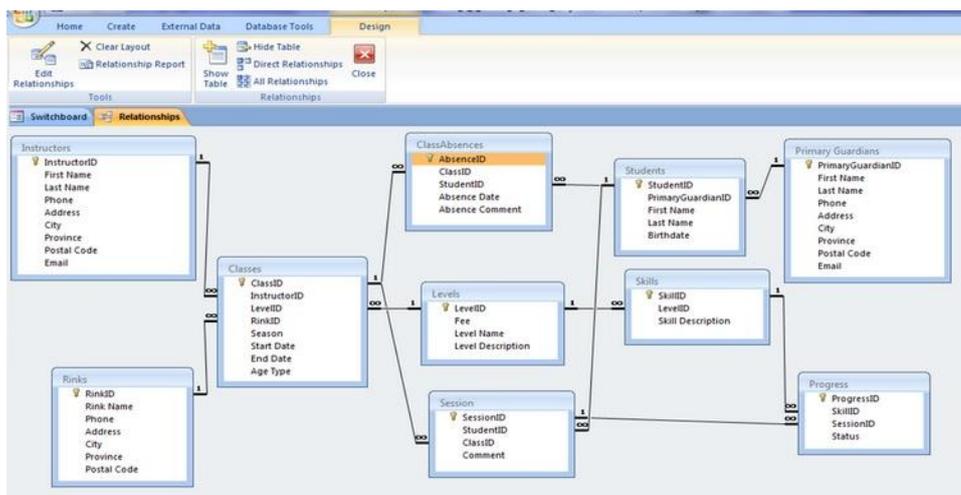
Microsoft Access salva in un unico file di tutti gli elementi utilizzabili, che possono essere anche usati per lo sviluppo di applicazioni complete. Gli elementi che possono essere memorizzati nel file sono:

- Tabelle
- Query
- Maschere
- Report
- Macro
- Pagine
- Moduli

Access da comunque la possibilità di progettare applicazioni nelle quali è necessario che si mantenga la separazione fisica tra le tabelle di dati (Back-End) e i restanti elementi (Front-End), così da migliorare l'aspetto della robustezza dell'applicativo, ma anche la sua distribuzione e manutenzione se condivisa tra diversi utenti.

Tabelle:

sono i contenitori dove vengono memorizzati i dati; è disponibile una interfaccia grafica elementare per la definizione o la modifica delle proprietà dei campi, inclusa la definizione degli indici e della chiave primaria (che può essere basata su più campi). Access da soli messaggi di avvertimento nei casi in cui la modifica dei campi può comportare perdita irreparabile dei dati.

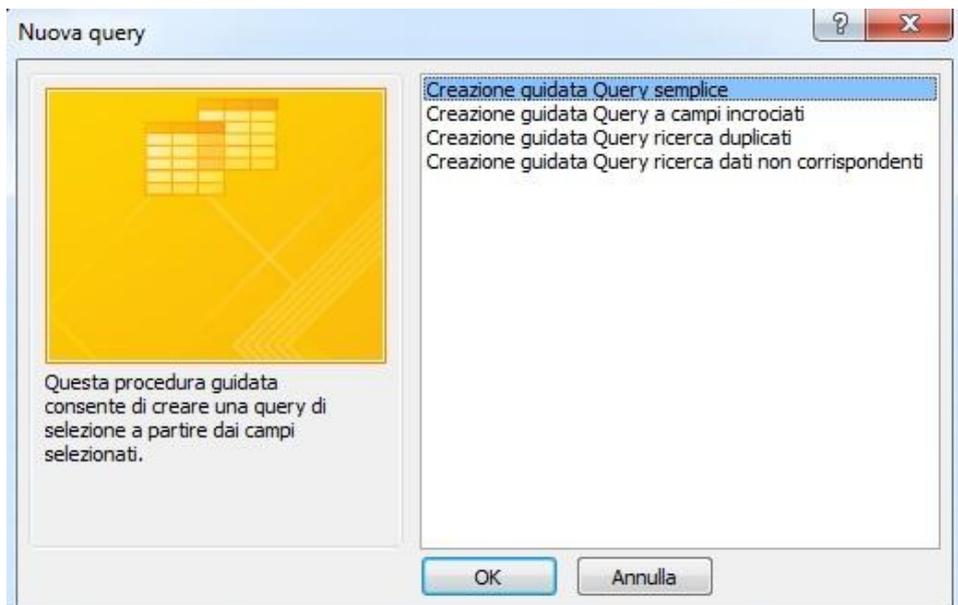


tabelle

collegate da relazioni

Query:

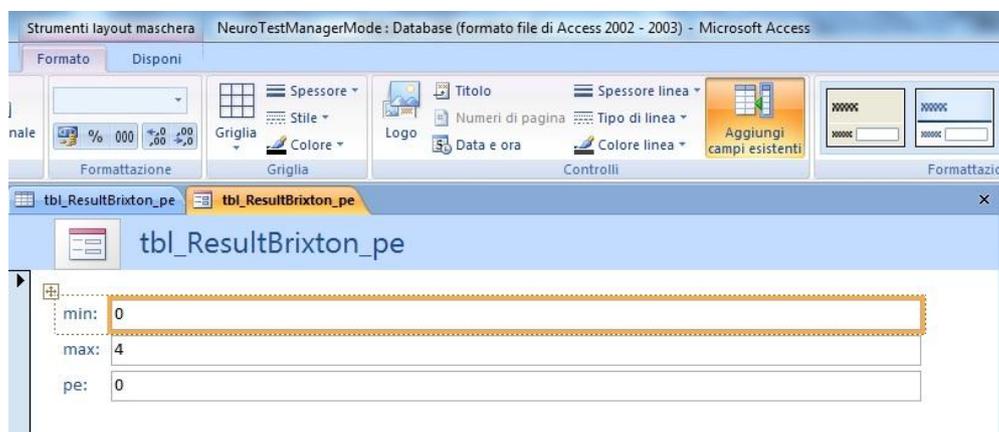
È lo strumento che serve per l'interrogazione e la manipolazione dei dati. Access dispone di un ambiente grafico per la definizione delle query (detto Query By Example o QBE) che permette anche ad utenti poco esperti la loro costruzione, con un minimo di controllo della correttezza sintattica; questa facilità, per contro, può comportare situazioni di blocco del sistema come conseguenza di errori concettuali.



*Creazione di una query guidata su Access 2007*

### Maschere:

Le maschere consistono negli elementi grafici utili alla interazione da parte degli utenti con i dati delle tabelle o delle query. Le maschere possono contenere gli elementi standard di Access ed elementi aggiuntivi. Le maschere possono includere codice VBA (visual basic application) che serve all'automazione degli elementi contenuti.



*Un layout di maschera*

### Report:

I report sono strumenti che servono alla visualizzazione, dei risultati basati sui dati, tabelle e query. Sono disponibili funzioni di base, quali aggregazione dei dati e totali parziali.

Macro:

Le macro sono oggetti che possono contenere una semplice sequenza di istruzioni, questi oggetti, per loro natura hanno una scarsa interazione con l'utente, ma la loro costruzione è semplice e si dividono in due parti:

- Pagine
- Moduli

*Pagine:*

Le pagine di accesso ai dati, che permettono la pubblicazione attraverso un web server.

*Moduli:*

Possono contenere moduli di codice e classi che si vuole rendere globali, e quindi richiamabili da un qualsiasi altro elemento dell' applicazione.

## - Breve descrizione di SQL server 2008:



Microsoft SQL Server è un DBMS relazionale (*Relational Database Management System* RDBMS), prodotto da Microsoft. Nelle prime versioni era utilizzato per basi di dati di grandi dimensioni. In questa versione i punti di forza delle componenti del sistema sono distinguibili in queste categorie:

- Gestione
- Scalabilità
- Prestazioni
- Sicurezza
- Sviluppo

### Gestione:

La gestione delle basi di dati di SQL Server avvengono attraverso la manipolazione delle *policy*, che sono un insieme di regole che possono essere stabilite sul server per fare in modo che tutti i dispositivi collegati (tablet, pc, workstation, server) siano gestite in maniera uniforme e abbiano tutti le stesse caratteristiche, le policy hanno anche in un' ampio grado di granularità, che consente di stabilire permessi e regole sulla base di dati in maniera dettagliata. (Ad esempio regole diverse per lo stesso utente in base al tipo di database acceduto oppure in base al tipo di account registrato su Windows Server.) È presente il meccanismo data collector che facilita la raccolta dei dati di gestione provenienti dalle diverse macchine consentendo anche la gestione degli errori.

### Scalabilità:

SQL Server 2008 introduce alcune nuove caratteristiche che semplificano queste operazioni ed in particolare mette a disposizione una funzionalità di compressione predefinita dei file dei database e dei file relativi ai transaction log. È possibile anche scegliere il livello di compressione (a livello di riga o di pagina). Questi tipi di compressione riducono la quantità di spazio richiesto e la quantità

di memoria necessaria a mantenere i dati, con i relativi ovvi benefici. SQL Server 2008 ha anche la compressione a livello di backup. È presente il Resource Governor (gestore delle risorse). Questa funzionalità serve a gestire il carico di lavoro e l'utilizzo delle risorse di sistema da parte di SQL Server, consentendo di specificare limiti sulla quantità di CPU e memoria che le richieste dell'applicazione in ingresso possono utilizzare. Ciò consente al sistema di lavorare in modo efficiente anche a partire da server di tipo entry level (non molto potenti e quindi nemmeno troppo costosi).

Sicurezza:

In SQL Server 2008 tale aspetto è migliorato con l'introduzione di due nuove caratteristiche: Extensible Key Management e Transparent Data Encryption. Il primo strumento fornisce opportuni meccanismi per memorizzare in modo sicuro le chiavi utilizzate nell'infrastruttura di crittografia. Il secondo offre una maggiore flessibilità nella crittografia dei dati, rendendo la crittografia una proprietà del database.

Sviluppo:

SQL Server 2008 fornisce una grande varietà di nuove caratteristiche atte a migliorare e facilitare il lavoro dei programmatori di database.

A screenshot of a SQL query editor window titled "SQLQuery3.sql - Daniele-PC\... (53)\*". The query text is: 

```
/****** Script per comando Seleziona le prime N righe da SSMS *****/  
SELECT TOP 1000 [min]  
                , [max]  
                , [pc]  
FROM [Neuro].[dbo].[tbl_ResultBrixton]
```

Una query SQL

Sono stati introdotti vari miglioramenti al linguaggio T-SQL. Un esempio concreto è l'inserimento nel sistema dell'istruzione *merge* che permette di verificare l'esistenza dei dati prima di inserirli in tabella. Inoltre non è più necessario creare join complessi per verificare l'esistenza dei dati al fine di effettuare un aggiornamento degli stessi o un nuovo inserimento. A livello di tipi di dati sono stati separati i tipi *date* e *time* con l'introduzione di due nuovi tipi di dati e sono stati introdotti due nuovi tipi di dati spaziali: *geography* e *geometry*. Un altro nuovo tipo di dati inserito in SQL server 2008 è il *filestream*, introdotto per risolvere il problema della memorizzazione e dell'utilizzo di oggetti binari di dimensioni consistenti, come documenti e file multimediali. Con il nuovo tipo



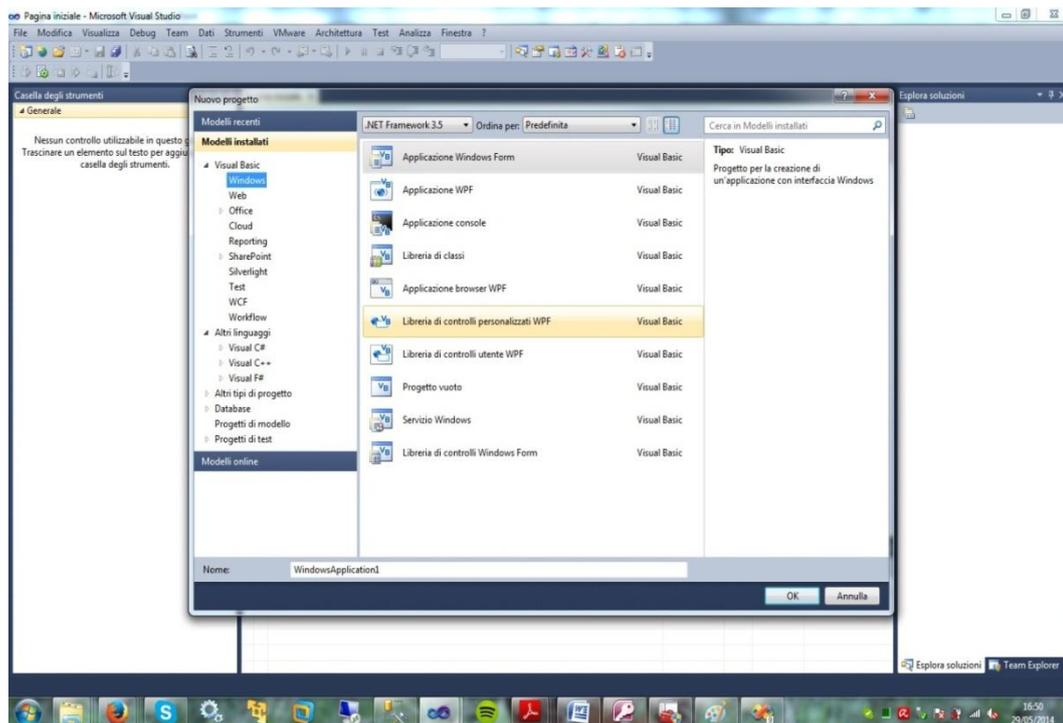
in diverse posizioni a utenti remoti o mobili tramite reti LAN o WAN, connessioni remote, connessioni wireless e Internet.

- *Reporting Services*: Reporting Services offre funzionalità Web aziendali per la gestione e la creazione di report utilizzando come contenuto una serie di origini di dati diverse, la pubblicazione dei report in vari formati e la gestione centralizzata della protezione e delle sottoscrizioni.
- *SQL Server Service Broker*: Service Broker consente agli sviluppatori la creazione di applicazioni di database protette e scalabili. La nuova tecnologia del Motore di database costituisce una piattaforma di comunicazione basata su messaggi che consente ai componenti di applicazioni indipendenti di agire come un insieme funzionalmente integrato. Service Broker include l'infrastruttura per la programmazione asincrona che è possibile utilizzare per le applicazioni all'interno di un unico database o di un'unica istanza, nonché per le applicazioni distribuite.

#### Differenze tra SQL server e Access:

Microsoft Access è un buon database per la gestione di piccole realtà, ma ovviamente ha i suoi limiti rispetto ad un database SQL Server. Per quanto riguarda la capacità ogni file Access ha una dimensione massima di 2GB, meno lo spazio necessario per gli oggetti contenuti in esso. Per aggirare questo problema si può lavorare con file multipli, ma quando la quantità di dati è nell'ordine di decine o centinaia di GB si comprende come tale tipo di gestione non sia adeguata. SQL Server invece consente di gestire efficientemente insiemi di dati di qualsiasi dimensione. Un altro aspetto da considerare è che quando si desidera modificare un file Access esso deve essere aperto esclusivamente da un solo utente. Questo significa che tutti gli altri potenziali utenti (si pensi ad un'azienda) non possono accedere ai dati finché non vengono apportate le modifiche. SQL Server elimina completamente questa limitazione e consente agli utenti di operare simultaneamente sullo stesso database. Infine per completare il confronto tra i due sistemi si all'accessibilità al database da internet. Con Access non è possibile creare link ai file di un database su un server remoto accessibile da internet, mentre questo è uno dei punti di forza di SQL Server.

## - Breve descrizione di Visual Studio 2008:



Schermata principale di Visual Studio.

Visual Studio, nella fattispecie la versione 2008, è il tool di sviluppo Microsoft per lo sviluppo su piattaforme Microsoft (Windows XP, Windows 7, Windows 8, Windows 8.1, Windows Server). Visual Studio 2008 utilizza il framework .NET in versione 3.5, e l' utilizzo di tale framework consente di sviluppare, tramite visual studio, un' ampia tipologia di applicazioni, che vanno dalle applicazioni desktop (Windows Form Application) alle applicazioni web (ASP.NET), servizi web ( Web Services), flussi di lavoro, componenti per Microsoft office. Inoltre Visual Studio supporta diversi linguaggi di programmazione: Visual Basic .NET, C#, C++, F#. Di fatto la prima cosa che Visual Studio presenta alla creazione di un nuovo progetto, è il tipo di applicativo che si vuole sviluppare e il linguaggio che si desidera adoperare. All' apertura di un nuovo progetto, Visual Studio si compone delle seguenti parti:

- Barra degli strumenti
- Casella degli strumenti
- Designer e Editor
- Solution Explorer
- Proprietà

- Elenco errori

### Barra degli strumenti:

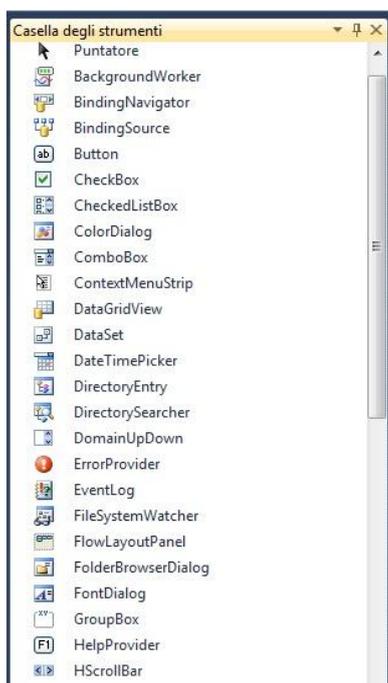
La barra degli strumenti fornisce un collegamento veloce alle funzionalità più comuni per il salvataggio dei singoli file o dell'intero progetto, e il debug del codice.



*barra degli strumenti*

### Casella degli strumenti:

La Casella strumenti (toolbox) è la finestra in cui sono posti tutti i controlli o gli oggetti che possono essere aggiunti ai contenuti trascinati nella finestra di editing sia in visualizzazione design che in visualizzazione codice. Gli oggetti presenti in questa finestra cambiano a seconda del tipo di file che si sta utilizzando nell'editor. Se stiamo lavorando su una Windows Form, ad esempio, troviamo i controlli per un'applicazione Windows.

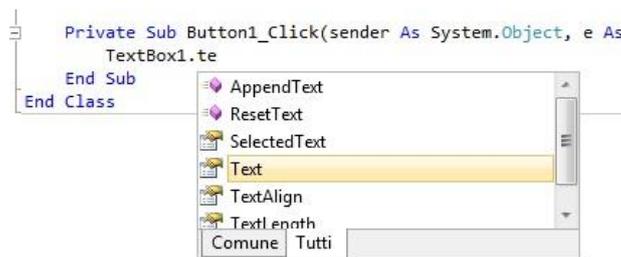


*Alcuni degli oggetti utilizzabili ( Windows Form Application)*

## Designer ed Editor:

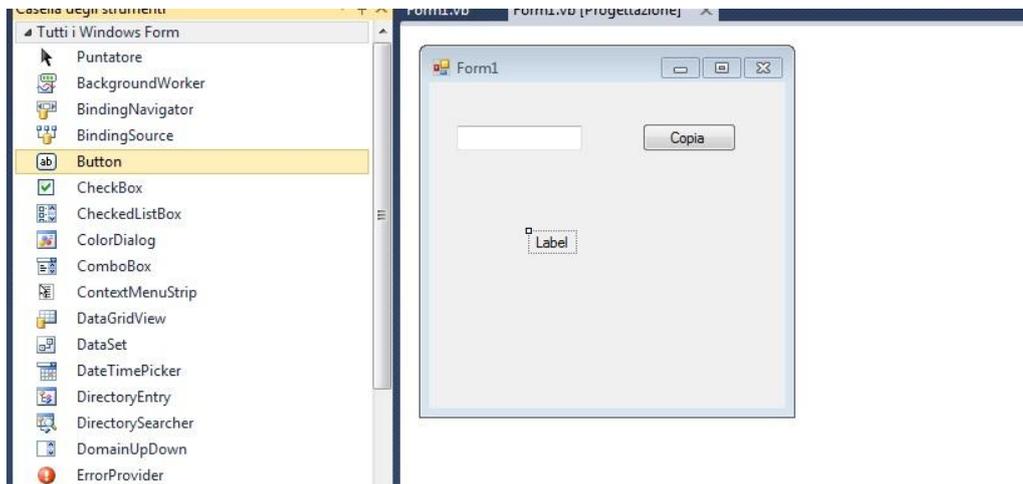
Visual Studio .NET offre un certo numero di designer visuali per il supporto allo sviluppo con le principali tecnologie legate al .NET Framework. Il designer di nostro interesse è quello per le Windows Application.

L' editor del codice è la parte centrale dell'IDE e fornisce funzionalità comuni a tutti i designer forniti in Visual Studio. Una delle più importanti è l'*intellisense*. Intellisense è un meccanismo che supporta il programmatore durante la stesura del codice delle applicazioni tramite un menu contestuale che in base alla posizione corrente del cursore presenta all'utente le possibili soluzioni da attuare.

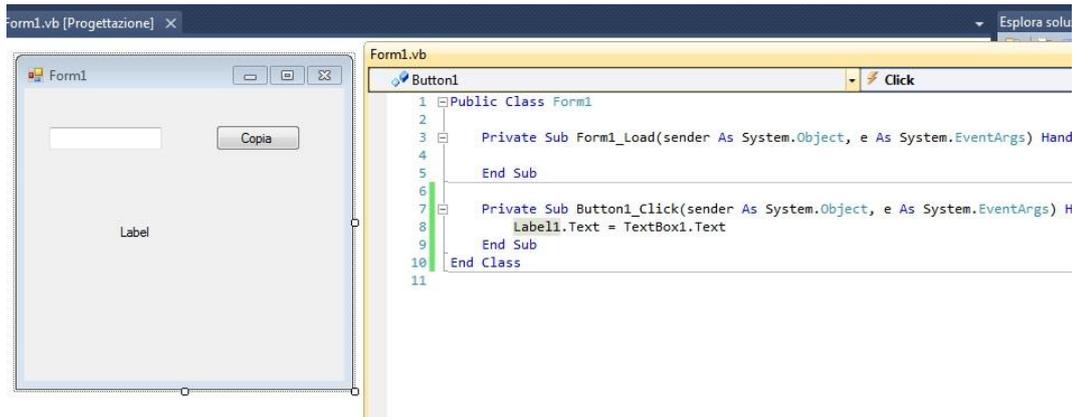


*Intellisense elenca proprietà e metodi utilizzabili nel caso della textbox*

Intellisense è fondamentale, in quanto evita la lettura continua della documentazione delle classi del .NET Framework da utilizzare, proponendo la lista completa di tipi, proprietà, metodi ed eventi per le classi dichiarate all'interno dell'applicazione e per quelle degli assembly esterni referenziati al progetto. Un'altra caratteristica importante del designer è che semplifica la programmazione degli oggetti, il programmatore non deve occuparsi della programmazione grafica del software in quanto è sufficiente che trascini (con il drag and drop) i controlli, dalla casella strumenti all'interno della form posizionandoli su di essa in un'area prescelta e Visual Studio genererà il codice corrispondente a quella impostazione grafica, perciò il programmatore dovrà occuparsi solo del core del software.



*Semplice form con tre controlli standard.*

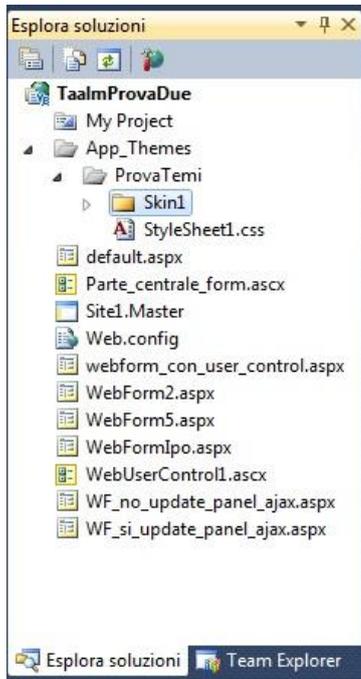


*Visualizzazione del codice e del design grafico.*

Il designer consente comunque di gestire anche gli aspetti automatizzati, che sono stati accennati in precedenza, in maniera arbitraria tramite l'editor di codice.

**Solution Explorer:**

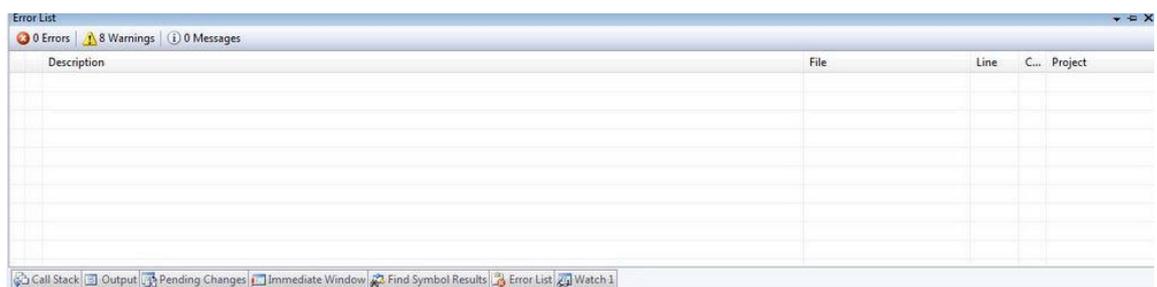
La finestra Solution Explorer è lo strumento principale per visualizzare e manipolare le soluzioni ed i progetti. Essa fornisce una semplice ma molto utile vista degli elementi presenti nella soluzione o nel progetto corrente e vi permette di interagire con essi direttamente mediante menù contestuali. Una soluzione è un "contenitore di progetti", che possono essere legati tra loro attraverso precisi collegamenti (references), in modo tale da poter comporre in un'unica applicazione le funzionalità sviluppate in livelli differenti.



*Solution Explorer: visualizza tutti i file del progetto.*

Elenco errori:

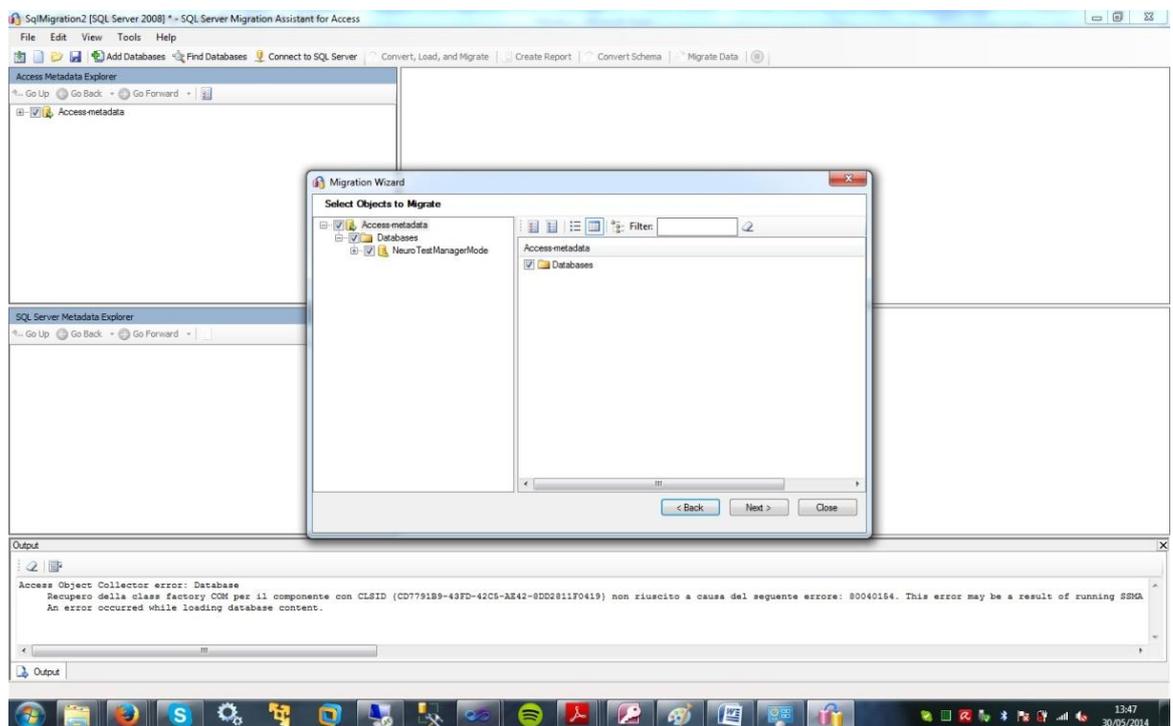
Questa finestra mostra l'elenco degli errori, dei warning e dei messaggi dati da una compilazione del progetto corrente. All'interno della tab "Errors" sono presenti gli errori di compilazione che non permettono la generazione degli assembly del programma. I warning sono invece degli avvisi, che permettono comunque la compilazione; esempi di warning sono l'utilizzo di classi marcate come obsolete, oppure, nel caso dell' HTML, dei tag non validi.



*finestra di debug di Visual Studio*

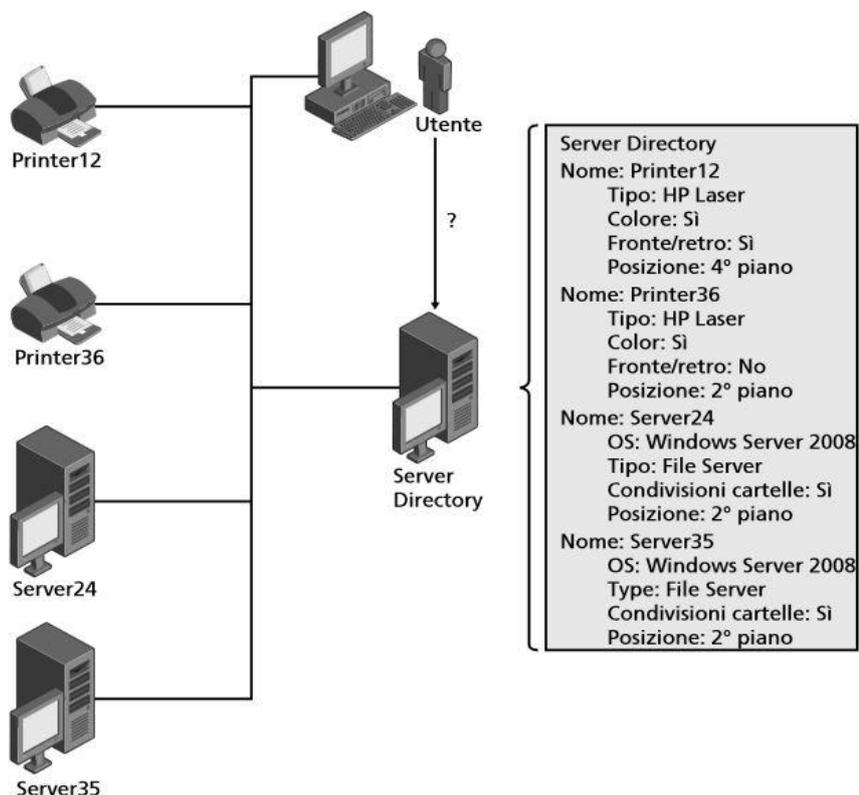
## -Breve descrizione di SQL migration assistant for access:

Microsoft SQL Server Migration Assistant (SSMA) per Access è uno strumento per la migrazione di database di Microsoft Access versioni 97, fino al 2010 a Microsoft SQL Server 2005 / Microsoft SQL Server 2008 / Microsoft SQL Server 2012 / Microsoft SQL Azure. Nel nostro caso il file di database di access viene migrato su SQL Server 2008. SSMA è dotato di un apposito wizard dove vengono illustrati i passaggi step by step per la migrazione.



*Wizard SSMA, seleziona il database da migrare.*

## -Breve descrizione di Active Directory:



### *Un' esempio di struttura Active directory*

Active Directory è un servizio di directory estendibile che si fonda sui concetti di dominio e di directory, quindi consente una gestione centralizzata delle risorse di rete (ad esempio cartelle e stampanti condivise), degli account utenti dei gruppi e dei computer. Tramite il paradigma SSO, Single Sign On, che prevede un controllo di accesso che consente di effettuare un'unica autenticazione valida per tutti i servizi e tutte le risorse del sistema. Active Directory, consente ad un qualsiasi utente registrato, il login da qualsiasi macchina o dispositivo presente nel dominio, e accedere a tutti i servizi e risorse disponibili, senza dover effettuare un login per ciascuno di essi. Il protocollo usato da active directory è LDAP, che è un protocollo standard per l'interrogazione per l'interrogazione e la modifica dei servizi di directory, e viene usato come base di dati che memorizza in forma centralizzata tutte le informazioni di autenticazione e accesso ai servizi, dando il vantaggio di mantenere tutte le informazioni sincronizzate tra i vari server di autenticazione di accesso alla rete.

### -Struttura di Active Directory:

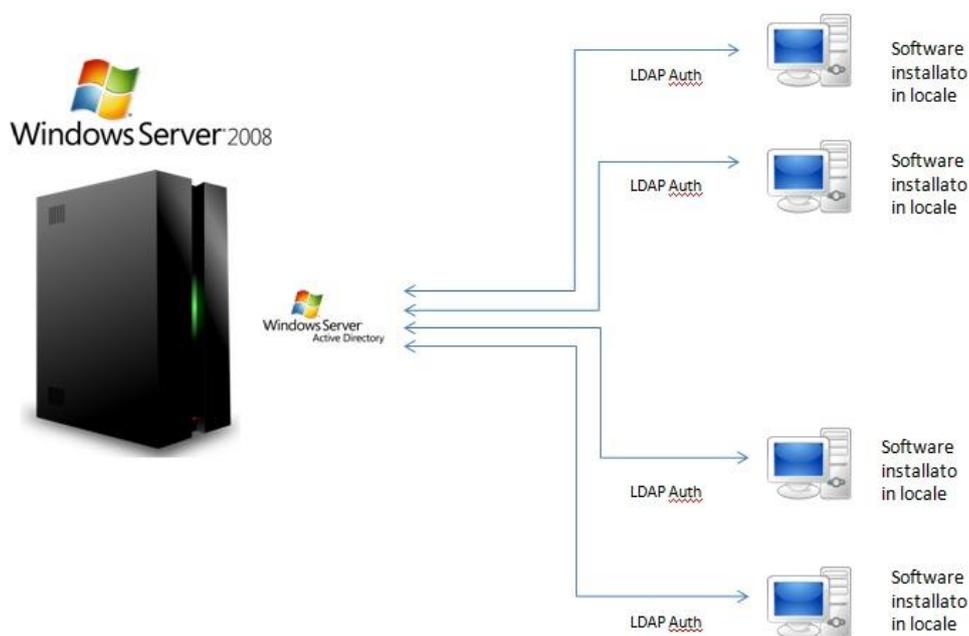
La struttura di active directory è suddivisa in Oggetti, Siti, Domini e Unità Organizzative.

- *Oggetti*: gli oggetti si suddividono in tre categorie: risorse, servizi e utenti. Ciascun oggetto rappresenta una singola entità (utente, computer, gruppo), e al suo interno può contenere altri oggetti, (gli attributi). L' oggetto è identificato univocamente dal suo nome, è composto da una serie di attributi, che sono definiti da uno schema. Lo schema, è esso stesso visto come un oggetto e ciò consente allo schema di essere esteso o modificato quando necessario. La modifica di uno schema, modifica la struttura di Active Directory, propagando la modifica su tutti i server di dominio Active Directory.
- *Siti*: Un Sito in Active Directory, è anch'esso un oggetto e rappresenta una locazione geografica fisica che ospita reti del dominio. I Siti contengono oggetti detti Sottoreti. I Siti possono essere usati per assegnare oggetti "politica di gruppo", semplificare l'individuazione delle risorse, gestire la replicazione della active directory, e gestire il traffico di collegamento alla rete. I Siti possono essere collegati ad altri Siti. Agli oggetti di un sito collegato possono essere assegnati costi che rappresentano la velocità, affidabilità, disponibilità o altre proprietà reali di una risorsa fisica.
- *Domini, alberi, foreste*: La struttura di Active Directory può essere suddivisa logicamente in tre diverse entità: i domini, gli alberi e le foreste. Un dominio rappresenta un insieme di macchine connesse fra di loro e che condividono un directory database comune, in cui sono inseriti gli oggetti. I domini sono identificati in base alla struttura del loro nome DNS, il namespace. Un albero è l'insieme di uno o più domini che condividono uno spazio di nomi contiguo. Tali domini sono collegati fra loro in modo gerarchico e i diversi controllori di dominio possono scambiarsi informazioni reciprocamente (transitive trust relationship). Al livello più alto della struttura viene definita la foresta, ovvero l'insieme di alberi presenti nella directory. Questi alberi condividono fra loro un catalogo globale, uno schema di directory, una struttura logica ed una configurazione. La foresta rappresenta perciò l'area in cui utenti, computer, gruppi ed altri oggetti sono accessibili.
- *Unità Organizzative*: Gli oggetti Active Directory presenti all'interno di un dominio possono essere raggruppati fra loro in unità organizzative (Organizational Units - OUs). Utilizzando queste strutture è possibile formare una gerarchia all'interno del dominio e

facilitarne così l'amministrazione, ad esempio suddividendo la struttura in termini geografici. Le unità organizzative sono il livello raccomandato per realizzare politiche di gestione dei gruppi, i quali sono oggetti Active Directory chiamati ufficialmente Group Policy Objects (GPOs), e per la delegazione di poteri amministrativi. Le unità organizzative rappresentano in ogni caso una semplice astrazione realizzabile per scopi amministrativi e non forniscono perciò un contenitore fisico di oggetti. In questo contesto non è possibile definire ad esempio un account utente con lo stesso nome in due unità organizzative appartenenti allo stesso dominio, poiché la visibilità effettiva è sempre limitata al dominio prescelto e non alle unità organizzative in esso contenute.

## - Breve descrizione di Windows Server 2008 r2:

Windows Server 2008 R2 è il sistema Windows in versione server usato per ospitare gli account di accesso al software tramite i servizi di active directory.



*schema di richiesta login*

Il software sarà installato localmente sui desktop, gli stessi fanno parte del dominio configurato opportunamente tramite active directory all'interno di Windows Server, in modo tale da centralizzare gli accessi. L'utente quindi dovrà avere un account utente su Active Directory quindi sul server. Una volta autenticato dal server avrà l'accesso all'uso del software.

## - Breve descrizione dell' interop per word ed access per l'esportazione dati:

Per avere un report diagnostico, quindi un referto anche stampabile su carta, si è rivelato necessario usare due applicazioni del pacchetto office.

- *Importare lo spazio dei nomi:* per utilizzare Microsoft Word, occorre importare nel progetto il riferimento a Microsoft Word, dal menu progetto, selezionare la voce aggiungi riferimento, nella finestra che viene aperta, selezionare la linguetta COM e la voce Microsoft Word 11.0 Object Library, va ricordato che il numero fa riferimento alla versione installata di Word.

codice per l' importazione in VB

```
Imports Microsoft.Office.Interop
```

- *Gli oggetti da utilizzare:* gli oggetti da utilizzare per interagire con Microsoft word, sono Word.application, che rappresenta l'applicazione word e Word.document che rappresenta il singolo documento. Di seguito si riporta la dichiarazione di tali oggetti

### **VB.Net**

```
'creo un oggetto di tipo word application
Dim objWord As Word.Application

'instanzio la classe
objWord = New Word.Application

'Rendo visibile word
objWord.Visible = True

'creo un nuovo documento
Dim objDoc As Word.Document

'aggiungo un documento
objDoc = objWord.Documents.Add

'rendo attivo il documento appena creato
objDoc.Activate()
```

Questo listato di codice apre un nuovo documento word con un template prestabilito per il referto medico, e all' interno del template sono stati inseriti dei campi chiamati bookmarks, che vengono riempiti dal software in base ai contenuti inseriti dall' utente.

L' interop di access è stato utilizzato per poter esportare ed elaborare a scopo statistico i dati provenienti dalle ricerche. (Ad esempio quanti pazienti nell' anno corrente hanno la malattia di Alzheimer).

codice per l'esportazione in un file access:

```
Vb net

If dgvResearch.Datagrid.Rows.Count = 0 Then
    MsgBox(angry)" Non esitono elementi per
l'esportazione ")
    Exit Sub
Else
    Dim path As String = ""
    sfdSaveDBForExport.FileName =
"testDB.mdb"
    sfdSaveDBForExport.ShowDialog()
    path =
IO.Path.GetFullPath(sfdSaveDBForExport.FileName)
    CreateAccessDatabase(path)

    Dim tableName As String = InputBo(angry)"
Inserisci il nome della tabella da salvare ", " ")
    If tableName = "" Then
        MsgBox("Non è stata salvata alcuna
tabella")
        Exit Sub
    End If

    Dim listOfNameOfCreatedTables As New
List(Of String)
    Dim columnCollection As
Data.DataColumnCollection = mViewToTable.Columns
    Dim filedString As New List(Of String)

    For Each columns As Data.DataColumn In
columnCollection
        Dim nameColumn As String =
columns.ColumnName.ToString
        filedString.Add("[ " & nameColumn & "]
TEXT(255) ")
    Next

    Dim arrayOfField As String() =
filedString.ToArray
```

```

        Dim concatenatedFields As String =
String.Join(" , ", arrayOfField)
        Dim researchData As DbData =
NtDbConnection.ResearchDataToExport(path)

        Dim researchDataTable As DbTable =
researchData.CreateTable(tableName,
concatenatedFields)
        For Each row As DataRow In
mViewToTable.Rows

researchDataTable.AddRow(row.ItemArray)
        Next
    End If

```

*Questo listato di codice serve per creare, un database, la tabella e le relative colonne.*

## - dll aziendale per controlli custom basata su controlli standard di sistema:

In azienda sono stati sviluppati dei controlli avanzati che sostanzialmente sono controlli nativi del framework DOT.NET customizzati al fine di garantire una miglior interfaccia utente e una migliore gestione dei dati. A livello grafico si può subito intuire la differenza tra un controllo custom e uno di sistema, in particolar modo la possibilità di poter inserire una descrizione e ad esempio un' unità di misura, senza dover aggiungere una label ad un controllo combo box nativo; semplificando così la composizione dell' interfaccia garantendo omogeneità e proporzioni. La caratteristica più importante è la gestione dei dati. Nella maggior parte dei software di archiviazione si deve far fronte al problema dei così detti "campi vuoti": usando i controlli nativi di sistema, qualora l'utente non inserisce o seleziona alcun tipo di dato al momento dell' immissione, si creano dei "buchi" nella base di dati chiamati DB NULL che creano inconsistenza nei dati stessi che portano la struttura della base di dati a essere instabile e quindi di generare errori. Questa problematica è stata risolta programmando i controlli in modo tale da far ritornare valore zero, nel caso dei tipi numerici (es: integer,float, long , int etc...), e stringa vuota nel caso dei tipi stringa. Di notevole importanza è un'altra particolarità di questa libreria , ovvero quella di fornire dei controlli customizzati anche per la gestione dei datagrid. Il controllo DataGridView nativo di Windows fornisce una tabella personalizzabile per la visualizzazione dei dati. La classe DataGridView consente la personalizzazione di celle, righe, colonne e bordi tramite l'utilizzo di proprietà . Queste proprietà sono state "riviste" per poter

migliorare ancor di più il concetto di personalizzazione e semplificare lato codice la creazione di tali viste.

## **- dll aziendale per semplificare l' interazione basi di dati/applicazione:**

La seconda dll utilizzata è di notevole importanza per un applicativo di questo genere, la peculiarità di questa libreria è quella di semplificare notevolmente l' interazione tra il software e la base di dati, attraverso un set di funzioni che permettono di effettuare le principali operazioni di inserimento eliminazione update e modifica etc... senza dover scrivere ogni volta la intera query SQL, evitando così errori sintattici e di struttura della base di dati. Le funzioni di questa libreria permettono di operare in modo corretto sia con access che con SQL server, infatti nonostante in corso di sviluppo, nella fase finale l'esportazione in SQL server della base di dati creata con access non ha costretto al team di programmazione, a dover effettuare alcuna modifica sul codice. Concludendo l' utilizzo di queste dll ha portato notevoli vantaggi in termini di coerenza del codice scritto e velocità di stesura.

## Capitolo 2: CONFIGURAZIONE E SVILUPPO DEL SOFTWARE:

-VISUAL STUDIO 2008 ( 64 CON IL PACCHETTO PER VISUAL BASIC )

I requisiti variano in base alle diverse combinazioni di componenti all'interno di Visual Studio 2008.

Per l'installazione occorrono:

- Computer con processore da 1,6 GHz o superiore
- RAM da 384 MB o superiore; per Windows Vista, 768 MB o superiore
- 2,2 GB di spazio disponibile su disco rigido
- Disco rigido da 5400 RPM
- Schermo con risoluzione 1024 x 768 o superiore
- Drive DVD-ROM

Visual Studio 2008 può essere installato sui seguenti sistemi operativi:

Tutte le edizioni di Windows Vista (x86 e x64) salvo la Starter Edition, Windows XP (x86 e x64) con Service Pack 2 o successivo (tutte le edizioni), Windows Server 2003 (x86 e x64) con Service Pack 1 o successivo (tutte le edizioni); Windows Server 2003 R2 (x86 e x64) successivo (tutte le edizioni).

-SQL SERVER 2008 R2 ( VERSIONE A 64 BIT )

- CPU Intel o compatibile a 1 GHz o più veloce.
- Almeno 512 MB di RAM.
- 2,2 GB di spazio disponibile su disco rigido.

I sistemi operativi supportati sono Windows 7, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Vista, Windows XP a 32 bit.

-OFFICE ULTIMATE 2007 :

- CPU Intel 500 megahertz (MHz) o superiore 1
- Almeno 256 megabyte (MB)
- 3GB di spazio disponibile sul disco rigido.
- Disco rigido da 5400 RPM
- Schermo con risoluzione 1024 x 768 o superiore
- Drive DVD-ROM

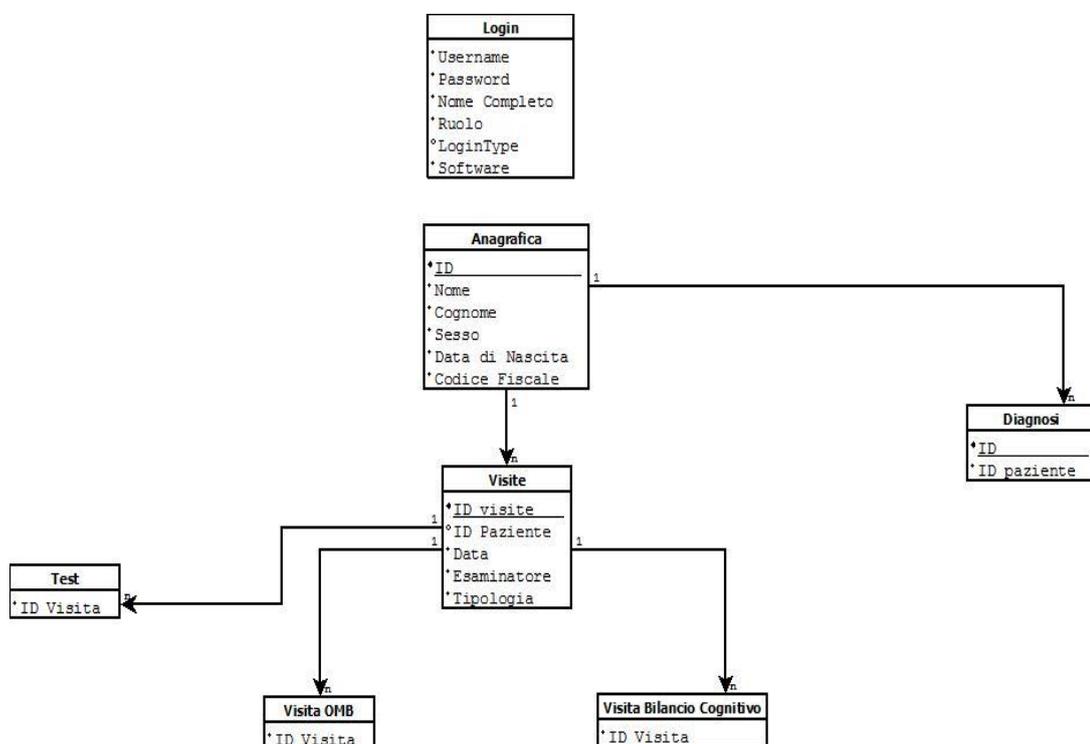
I sistemi operativi supportati sono Windows 7, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Vista, Windows XP a 32 bit.

WINDOWS 7 ULTIMATE:

- Processore a 32 bit (x86) o a 64 bit (x64) da 1 GHz o più veloce
- 1 GB di RAM (32 bit) o 2 GB di RAM (64 bit)
- 16 GB di spazio disponibile su disco rigido (32 bit) o 20 GB (64 bit)
- Scheda grafica DirectX 9 con driver WDDM 1.0 o superiore

## -Sviluppo db e struttura dati:

La base di dati è composta dalle tabelle Login, Anagrafica, Diagnosi, Test, Visite, Visite AMB, Visite B. Cognitivo.



Schema relazionale della base di dati

L'obiettivo è quello di rappresentare quindi una base di dati per la clinica neurologica, in modo che le seguenti informazioni siano opportunamente rappresentate:

- Un paziente viene nella clinica neurologica per effettuare una visita e viene registrato con i suoi dati anagrafici.
- il paziente deve essere registrato inserendo obbligatoriamente le seguenti informazioni : nome , cognome , data di nascita , sesso, luogo di nascita e codice fiscale.

- Oltre ai dati che devono essere inseriti obbligatoriamente l'utente può inserire anche altri tipi di campi che servono a precisare meglio il profilo del paziente : numeri di telefono scolarità caregiver ecc...
- L'utente ( che in questo caso si presuppone sia un dottore o un suo collaboratore ) può effettuare una nuova visita ( ambulatoriale o bilancio cognitivo ) oppure stilare una diagnosi in base a dati clinici precedentemente verificati nella stessa clinica e quindi registrati già nel database oppure in base a documentazione cartacea proveniente da altri parti.
- La visita ambulatoriale il bilancio cognitivo e la diagnosi non sono strettamente legate tra loro , un paziente può essere sottoposto a più visite sia ambulatoriali che bilancio cognitivo prima di stilare una diagnosi e viceversa .
- Nel caso di una diagnosi al paziente possono essere legati le informazioni sulla provenienza sulla familiarità e in particolar modo sulla patologia che è il campo che maggiormente interessa l'utente e serve a delineare in modo concreto la diagnosi stessa.

Nel caso di una visita esistono due tipi di visite :

- Ambulatoriale
- Bilancio

La visita ambulatoriale prevederà l'inserimento di dati che corrispondono a grandi linee ai parametri vitali del paziente e comprende anche la possibilità di effettuare alcuni test riguardanti invece il bilancio cognitivo.

Il bilancio cognitivo sostanzialmente è una visita che raccoglie un ampio elenco di test che possono essere effettuati al paziente e che consentono di ottenere un quoziente di sviluppo intellettuale cioè distinguere tra normalità e ritardo mentale.

L'utente può accedere a secondo del tipo di visita che sta eseguendo ad un determinato elenco di test effettuabili .

Ogni test convalidato avrà un Id che sarà lo stesso della visita di riferimento in modo che sia determinato se mi trovo in ambito ambulatoriale o no e su ognuno c'è l'arbitrarietà di eseguirlo modificarlo o effettuarlo in un secondo momento.

## TABELLE:

Login :

	Nome campo	Tipo dati
	Password	Testo
	NomeCompleto	Testo
	Ruolo	Testo
	LoginType	Testo
	Software	Testo

*Login*

Anagrafica:

	Nome campo	Tipo dati
🔑	ID	Testo
	Nome	Testo
	Cognome	Testo
	Sesso	Testo
	DataDiNascita	Data/ora
	LuogoDiNascita	Testo
	CodiceFiscale	Testo
	ViaResidenza	Testo
	CAPResidenza	Testo
	LocalitàResidenza	Testo
	ProvinciaResidenza	Testo
	NazioneResidenza	Testo
	Professione	Testo
	Telefono	Testo
	TelefonoAlternativo	Testo

*Anagrafica*

Visite :

	Nome campo	Tipo dati
🔑	IDVisita	Testo
	IDPaziente	Testo
	Tipologia	Testo
	Data	Data/ora
	Esaminatore	Testo

*Visite*

Visita ambulatoriale :

Nome campo	Tipo dati
IDVisita	Testo
IDPaziente	Testo
Data	Testo
Terapie	Testo
PaSistolica	Testo
PaDiastolica	Testo
FC	Testo
Ritmo	Testo
Peso	Numerico
MMSE	Testo
ADL	Testo
IADL	Testo
ObiettivoNeurologico	Testo
Esaminatore	Testo

*Visita Ambulatoriale*

Vista bilancio cognitivo

La vista del bilancio cognitivo raccoglie un dato significativo per ogni test che per ognuno corrisponderà precisamente al punteggio definitivo che sancisce se il test ha avuto un esito positivo o no.

DATA	MMSE	MATRICI PROGRESSIVE DI RAVE	LURIA MOTOR SEQUENCE	STROOP TEST PARTE 1	STROOP TEST PARTE 2	BRIXTON
23/06/2014	0	---	---	---	Alterato	---

Test ( esempi di alcune delle tabelle )

Nome campo	Tipo dati
IDVisita	Testo
PunteggioGrezzo	Numerico
PunteggioCorretto	Numerico
PunteggioEquivalente	Numerico

Nome campo	Tipo dati
IDVisita	Testo
SubsetDiOrientamento	Numerico
SubsetDiRegistrazione	Numerico
SubsetDiAttenzioneEcalcolo	Numerico
SubsetDiRichiamo	Numerico
SubsetDiLinguaggio	Numerico
SubsetDiCopia	Numerico
PunteggioGrezzo	Numerico
PunteggioCorretto	Numerico

## -CONCETTO DI ASTRAZIONE DEI DATI

Nella progettazione di un sistema software è opportuno basarsi sul principio della modularità, affinché esso risulti più semplice da progettare, realizzare e mantenere.

I moduli sono componenti (parti) del sistema che realizzano un'astrazione.

I meccanismi di astrazione più diffusi sono:

1. astrazione sul controllo;
2. astrazione sui dati;

**Astrazione sul controllo:**

Astrazione di una funzionalità dai dettagli dell'implementazione.

E' ben supportata dai linguaggi di programmazione tradizionali tramite il concetto di sottoprogramma.

**Astrazione sui dati:**

Consiste nell'astrarre le entità (oggetti) costituenti il sistema, descritte in termini di una struttura dati e delle operazioni possibili su di essa.

Può essere realizzata con un uso opportuno delle tecniche di programmazione modulare nei linguaggi tradizionali.

E' alla base della modellazione a oggetti.

E' supportata da appositi costrutti nei linguaggi di programmazione ad oggetti.

*Oggetti astratti:*

Un oggetto (astratto) è costituito da una struttura dati e dalle operazioni possibili su di essa.

Il valore corrente che assume la struttura dati costituisce lo stato dell'oggetto.

Tipi di dati astratti (TDA)

Un TDA è la specifica di un insieme di oggetti astratti.

Un TDA è un tipo d'utente, cioè non predefinito dal linguaggio.

Come tipo di dati, è una specifica, va istanziato per creare specifici oggetti.

Gli esemplari del tipo sono oggetti astratti.

*Tipi di dati astratti:*

Il concetto di tipo di dato in un linguaggio di programmazione tradizionale è quello di insieme dei valori che può assumere un dato (una variabile).

Il tipo di dati astratto (TDA) estende questa definizione, includendo anche l'insieme di tutte e sole le operazioni possibili su dati di quel tipo. La struttura dati "concreta" è incapsulata nelle operazioni su di essa definite.

Non è possibile accedere alla struttura dati incapsulata (né in lettura né in scrittura) se non attraverso le operazioni definite su di essa.

Un vantaggio: la struttura dati interna non può essere alterata da operazioni scorrette da parte dell'utente, in quanto ad essa si accede solo tramite le operazioni previste e realizzate dal produttore.

Interfaccia: specifica del TDA, descrive la parte direttamente accessibile dall'utilizzatore.

Cliente o utilizzatore fa uso del TDA per realizzare procedure di un'applicazione, o per costruire TDA più complessi.

Il produttore realizza le astrazioni e le funzionalità previste per il dato.

Un produttore di un TDA può essere utilizzatore di un altro TDA e una modifica nella sola realizzazione del TDA non influenza i moduli che ne fanno uso (in quanto non cambia l'interfaccia).

## -Sviluppo codice e gestione della struttura:

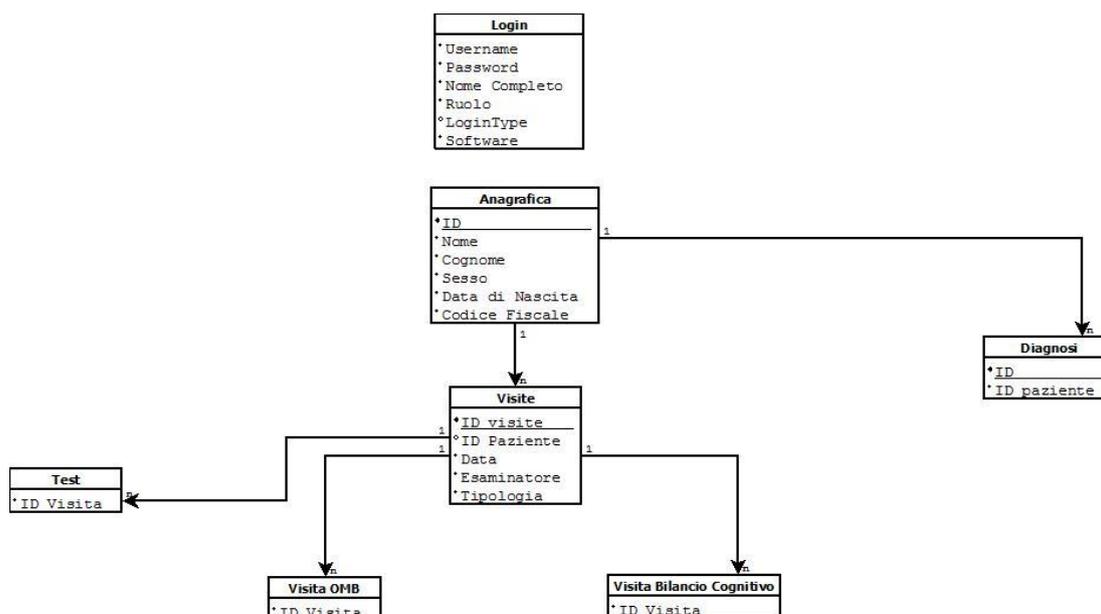
A fronte di questa analisi e per cercare di trarre più vantaggi possibili si è scelto di astrarre anche le relazioni che intercorrono tra le tabelle e fare in modo che anche queste rientrino nel paradigma TDA.

- 1) GESTIONE CHIAVE PRIMARIA/ESTERNA
- 2) GESTIONE RELAZIONI (query sql)
- 3) CREAZIONE DINAMICA CONTROLLI.

1) Nel contesto dei database relazionali, una chiave esterna (ingl. foreign key) è un vincolo di integrità referenziale tra due o più tabelle. Essa identifica una o più colonne di una tabella (referenziante) che riferenzia una o più colonne di un'altra tabella (referenziata).

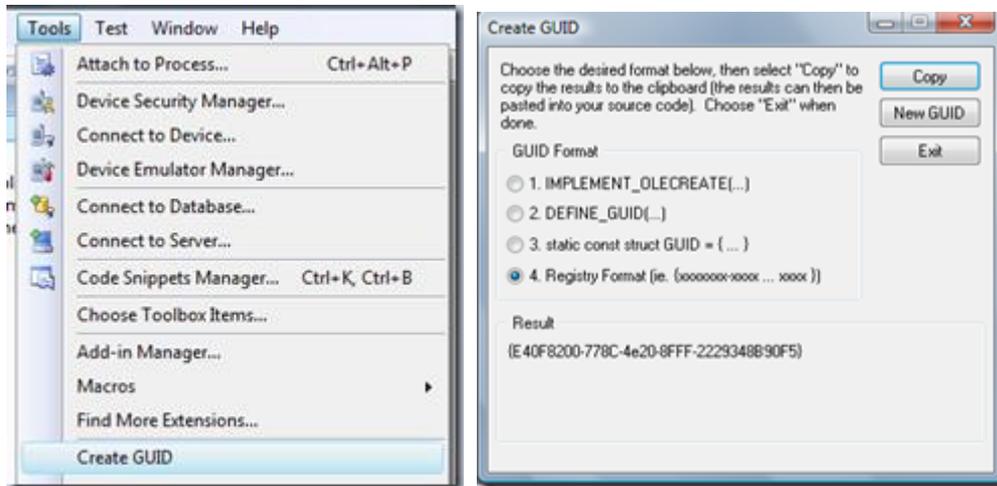
I valori di un record delle colonne referenzianti devono essere presenti in un unico record della tabella referenziata. Ciò implica che un record nella tabella referenziante non può contenere valori che non esistono nella tabella referenziata (eccetto nel caso particolare di valori NULL). Più record della tabella referenziante possono puntare allo stesso record della tabella referenziata. Nella maggior parte dei casi, ciò corrisponde alla relazione "uno a molti" tra una tabella padre ed una tabella figlio.

Riprendendo il seguente schema relazionale :



Schema della base di dati

è evidente dove sia necessario creare le chiavi primarie e le chiavi referenziali .  
Le chiavi referenziali non vengono fatte generare automaticamente dalla base di dati attraverso il contatore che genera numeri interi ma bensì vengono create sotto forma di stringhe alfa numeriche dal programma attraverso il tool *Create Guid*.

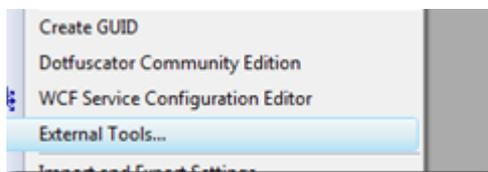


*Create GUID*

Come si può notare creare e copiare un nuovo GUID non è proprio un'operazione veloce perché va selezionata la tipologia -> generato il codice -> copiato ed infine premuto il tasto Exit per chiudere il tutto.

Indagando velocemente ho scoperto che c'è un modo più comodo e veloce ossia farsi generare un GUID nella finestra Output Window di Visual Studio 2008. Ecco come si fa:

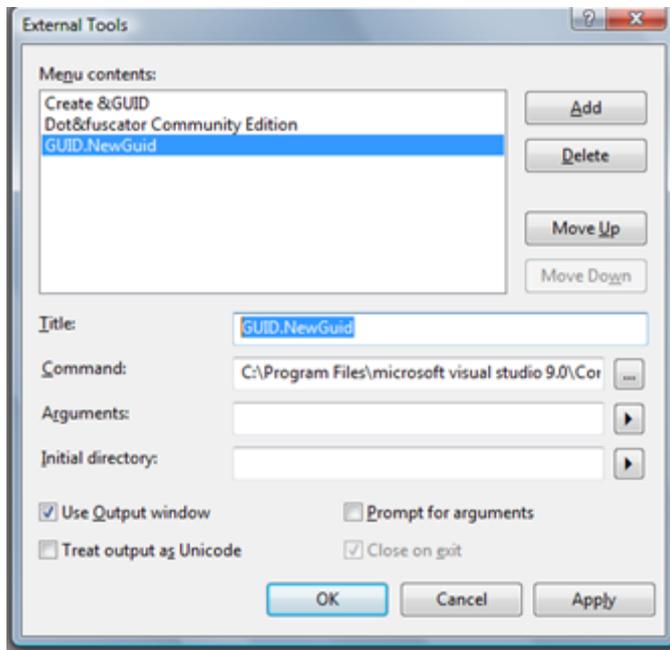
Aprire la finestra "External Tools..."



*Screenshot menu*

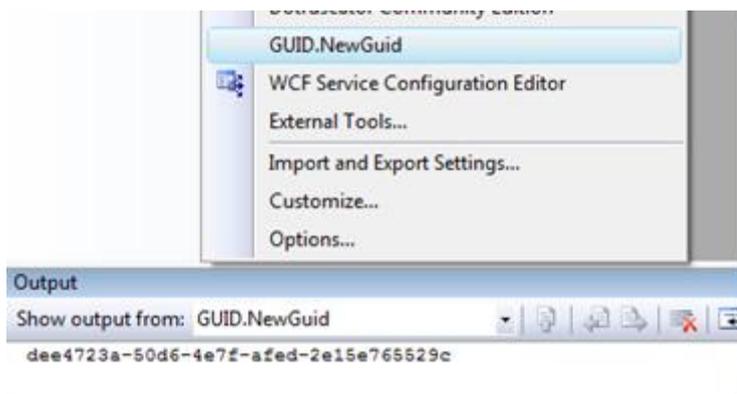
Aggiungere un nuovo tool che come command è impostato sull'eseguibile **uidgen.exe** che si trova all'interno della folder c:\[Programs Files]\Microsoft visual studio 9.0\Common\Tools.

Infine impostare un titolo a piacimento (es. GUID.NewGuid) e cosa molto importante selezionare *Use Output Window* che per l'appunto indica al tool di generare il codice nell'Output Window



Finestra di External Tools

A questo ogni volta che desideriamo generare un nuovo GUID basta selezionare il comando che abbiamo creato e quindi ci verrà creato un nuovo codice nell'Output Window.



finestra di Output

Ultima nota, se per qualche motivo abbiamo bisogno di un nuovo GUID al di fuori dell'ambiente di sviluppo lo si può generare in PowerShell con il seguente comando:

```
[System.Guid]::NewGuid().ToString()
```

Una volta stabilito come gestire e creare le chiavi referenziali sono entrate in gioco le dll utili alla gestione della banca dati. Le funzioni che ci interessano particolarmente sono quelle che ci semplificano le operazioni di inserimento eliminazione e modifica dei dati. Seguendo la struttura della base di dati e come l'utente interagisca con essa attraverso il sw , facciamo un esempio delle principali funzioni e di come queste funzioni vengano utilizzate confrontandole con la sintassi Sql equivalente. Ad esse viene applicato il concetto di overloading; in pratica una famiglia di funzioni/subroutine aventi lo stesso nome, ma con la possibilità di accettare un diverso set di argomenti (signature), ed eventualmente restituire un diverso valore di ritorno.

## -INSERIMENTO

```
Public Function AddRow(ByVal columnNames() As String, ByVal values() As Object) As Boolean
```

```
Public Function AddRow(ByVal values() As Object) As Boolean  
Member of DatabaseLink.DbTable
```

```
Public Function AddRow(ByVal columns() As System.Data.DataColumn,  
ByVal values() As Object) As Boolean Member of DatabaseLink.DbTable
```

Come è intuibile dalla sintassi la funzione AddRow serve a registrare una nuova riga nel database. Alla funzione devono essere passati un array di stringhe che contiene i nomi delle colonne relative alla tabella nella quale si vogliono inserire i dati e una array di valori di tipo Object ( possono essere inserite stringhe interi decimali valori booleani ecc...) . Se l'inserimento è andato a buon fine la funzione restituirà un valore booleano = True altrimenti False; ciò semplifica di molto la costruzione della stringa sql poichè viene automatizzata la sua costruzione attraverso l'istruzione Insert .

## Insert

Il comando ha la funzione di inserire i dati nelle tabelle.

Le colonne (o campi) di destinazione dei valori possono essere o meno dichiarate nel comando. Se non vengono dichiarate, è necessario passare al comando un valore per ogni colonna della tabella, rispettando rigorosamente l'ordine delle colonne stesse. Se, invece, le colonne di destinazione vengono dichiarate, è possibile indicare le sole colonne per le quali vengono passati dei valori, purché vengano inseriti comunque i valori per tutte le colonne not null (che non possono essere nulle) della tabella.

## Sintassi del comando insert

```
INSERT INTO nome_tabella  
VALUES (elenco valori, tutti, rispettando l'ordine dei campi  
della tabella, ad es "donato");
```

oppure

```
INSERT INTO nome_tabella (elenco dei campi interessati  
dall'inserimento)  
VALUES (elenco valori, tutti, rispettando l'ordine dei campi  
dichiarati sopra, esempio "donato");
```

## -MODIFICA

```
Public Function ModifyRow(ByVal row As System.Data.DataRow, ByVal  
modifiedValues() As Object) As Boolean
```

```
Public Function ModifyRow(ByVal row As System.Data.DataRow, ByVal  
columnNames() As String, ByVal modifiedValues() As Object) As  
Boolean
```

```
Public Function ModifyRow(ByVal row As System.Data.DataRow, ByVal  
columns() As System.Data.DataColumn, ByVal modifiedValues() As  
Object) As Boolean
```

```
Public Function ModifyRowValue(ByVal row As System.Data.DataRow,  
ByVal column As System.Data.DataColumn, ByVal modifiedValue As  
Object) As Boolean
```

La funzione modifyRow , serve a modificare righe o valori già presenti nel database . Riassumendo le caratteristiche di questa funzione , i dati principali che si devono aver a disposizione sono :

- il record ( la row o riga già presente nel database ma da modificare).
- Le colonne relative ai campi che si voglio modificare.
- I valori modificati che si vogliono sostituire.

Se l'operazione va a buon fine la funzione restituisce un valore booleano = True.

Per la costruzione della stringa Sql viene utilizzata l'istruzione update.

## Update

Il comando update ha la tripla funzione di modificare i dati delle tabelle.

Il nome di ogni campo che deve essere modificato va dichiarato dopo la parola chiave SET e deve essere seguito dal simbolo " = " (uguale) e dal nuovo valore che deve assumere.

È possibile modificare più campi della stessa riga in un unico comando update, separandoli l'uno dall'altro con il simbolo ", " (virgola).

Il comando generico aggiorna tutte le righe della tabella. È possibile restringerne il numero applicando la parola chiave aggiuntiva WHERE, che permette di effettuare una selezione qualitativa delle righe imponendo delle condizioni sui dati presenti nelle righe prima dell'aggiornamento.

Sintassi del comando update

- Update generico:

```
UPDATE nome_tabella
SET nome_campo1 = 'valore1_nuovo',
    nome_campo2 = 'valore2_nuovo',
    ...
;
```

- Update con condizione:

```
UPDATE nome_tabella
SET nome_campo1 = 'valore1_nuovo',
    nome_campo2 = 'valore2_nuovo'
WHERE nome_campo3 = 'valore';
```

A prima impressione al programmatore potrebbero sorgere dei dubbi su come reperire il record che si vuole modificare ma ciò risulta relativamente semplice perchè entra in gioco una funzione molto importante che viene utilizzata spesso all'interno del codice per poter effettuare altre operazioni sui record :

```
Public Function FindRow(ByVal values() As Object) As System.Data.DataRow
```

```
Public Function FindRow(ByVal columnName As String, ByVal value As Object) As System.Data.DataRow
```

La funzione restituisce un datarow cioè riporta proprio il tipo di dato che la precedente funzione Modify richiede. Per poter cercare il record interessato si consiglia di inserire il suo identificativo e il nome del campo nel quale è inserito ; più chiaramente :

ID
02849ea3-f669-4f0f-a119-2965cf41e45a
02c13ee7-8738-45eb-adc6-d472a044dd01
07fdc482-d992-46ac-a1a2-2441f6606dc6
0a7314fb-30c7-4b32-99e7-5f4016aefd76
0b986a15-a4cc-4238-a643-66bc764a1078
0d12da39-0504-44b3-9dfa-0acaa49bb5c5
0f2f2e2e-b30c-4ade-96cb-8eca68592fbc

Record dei GUID

Facendo riferimento a questa tabella se voglio avere a disposizione tutto il record con i relativi dati è sufficiente inserire il nome della colonna "ID" e il valore in questo caso espresso come Guid.

Per poter selezionare la riga questa integra l'istruzione Select

Col comando select abbiamo la possibilità di estrarre i dati, in modo mirato, dal database.

*Sintassi del comando select*

```
SELECT [ ALL | DISTINCT | TOP ] lista_elementi_selezione
FROM lista_riferimenti_tabella
[ WHERE espressione_condizionale ]
[ GROUP BY lista_colonne [HAVING Condizione] ]
[ ORDER BY lista_colonne ];
```

## Cancellazione

Il comando Select ritorna molto utile anche per un altro tipo di operazione che solitamente l'utente è solito effettuare: l'eliminazione .

```
Public Function DeleteRow (ByVal row As System.Data.DataRow) As Boolean
```

Infatti come è ben intuibile anche questa accetta come tipo di dato un Datarow e restituisce sempre un valore booleano utile a certificare l'esito positivo dell'operazione e costruisce la stringa Sql integrando l'istruzione delete.

## Delete

Il comando delete ha la funzione di cancellare i dati dalle tabelle. Come il comando update anche delete può operare in modo generico cancellando tutte le righe della tabella oppure può identificare le righe da cancellare mediante la parola chiave aggiuntiva WHERE e la condizione (o le condizioni) ad essa associata.

#### *Sintassi del comando delete*

- Delete generico:

```
DELETE FROM nome_tabella;
```

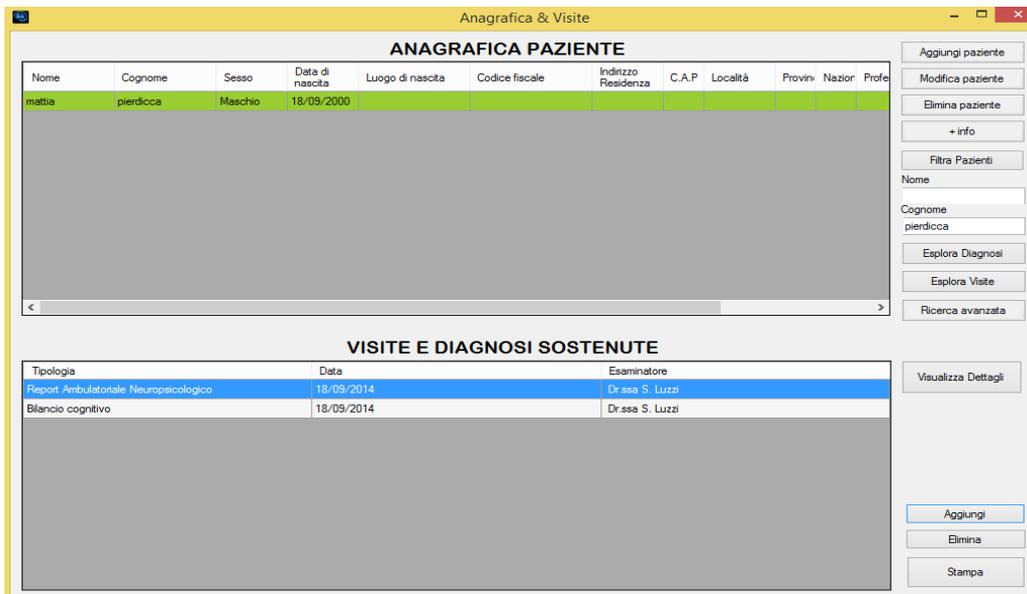
- Delete con condizione:

```
DELETE FROM nome_tabella  
WHERE nome_campo = 'valore';
```

Stabilito come devono essere create le chiavi e quali funzioni utilizzare per poter effettuare le operazioni principali, resta capire come gestire le relazioni. Per poterle spiegare al meglio è utile simulare un esempio di utilizzo del sw, in particolare l'aggiunta di una nuova visita.

Seguendo la logica strutturale quello di cui abbiamo bisogno è "portarci dietro" la chiave primaria dell'anagrafica (l'Id) e utilizzarla come reference Key nella visita. Ciò avviene quindi facendo passare alla funzione CreateRow l'identificativo del paziente come tagObject:

```
Public Function CreateRow(ByVal cellValues() As Object, ByVal  
tagObject As Object) As System.Windows.Forms.DataGridViewRow.
```



Schermata principale del software

Nella vista quindi una volta selezionata la riga relativa al paziente verrà prelevato il suo tag relativo attraverso la funzione e memorizzato in una variabile temporanea:

```
GetSelectedRowTag ()
```

Una volta selezionato il paziente e la visita che si vuole effettuare ; il tag ovvero l'id del paziente verrà incluso nei dati relativi alla visita sfruttando il costruttore.

Il costruttore, in informatica e specificamente nella programmazione orientata agli oggetti, indicano dei metodi di programmazione associati alle classi che hanno lo scopo di inizializzare le variabili di istanza.

```
Public Sub New(ByVal interactionMode As Boolean, ByVal
idSelectedVisita As String, ByVal idPaziente As String, ByVal
nameSurname As String)
    InitializeComponent()
    mStopEvents = False
    ' la variabile interaction aiuta a definire se ho cliccato
aggiungi o modifica
    mInteractionMode = interactionMode
    mSelectGuidVisita = idSelectedVisita
    mNewGuid = Guid.NewGuid.ToString
    mPatientId = idPaziente
    mTipologia = "Bilancio cognitivo"
    mNameSurname = nameSurname
    mJustCalculated = False
End Sub
```

Continuando nell'esempio d'uso proviamo ad effettuare una visita:

**ANAGRAFICA PAZIENTE**

Nome	Cognome	Sesso	Data di nascita	Luogo di nascita	Codice fiscale	Indirizzo Residenza	C.A.P.	Località	Provin	Nazion	Profe
mattia	pierdicca	Maschio	18/09/2000								

**frmSelectVisita**

Selezionare il tipo di visita da effettuare per : pierdicca mattia

Bilancio cognitivo  
Report Ambulatorio

Ok Annulla

**Tipologia**

Report Ambulatoriale Neuropsicologico
Bilancio cognitivo

18/09/2014 Dr.ssa S. Luzi

Inserimento visita

**Esegui i test**

**TEST** pierdicca mattia

Calcola Reset Convalida

Data : mercoledì 24 settembre 2014

Esaminatore

**Test**

- MATRICI PROGRESSIVE DI RAVEN
- RICONOSCIMENTO VOLTI FAMOSI
- LURIA MOTOR SEQUENCE
- STROOP TEST PARTE 1
- STROOP TEST PARTE 2
- BRIXTON
- DIGIT SPAN
- CUBI DI CORSI
- FIGURA A DI REY COPIA
- FIGURA A DI REY RIEVOCAZIONE
- FIGURA B DI REY COPIA
- FIGURA B DI REY RIEVOCAZIONE BREVE
- FIGURA B DI REY RIEVOCAZIONE LUNGO
- LISTA PAROLE DI REY BREVE TERMINE
- LISTA PAROLE DI REY LUNGO TERMINE
- PRASSIA IDEOMOTORIA MANO DOMINANTE
- PRASSIA IDEOMOTORIA MANO NON DOMINANTE
- PRASSIA BUCCO LINGUO FACCIALE
- FLUENZA PER CATEGORIE
- FLUENZA FONEMICA
- DENOMINAZIONE
- LETTURA
- MATCHING
- FIGURE SOVRAPPOSTE DI LG
- VOSP
- IADL
- VALUTAZIONE NEUROPSICHIATRICA
- ADL

**IADL**

- telefono
- spesa
- pasto
- casa
- bucato
- spostamento
- farmaci
- denaro
- IADL
- PercentIADL

Ok Annulla

Scelta Test

Quando verrà cliccato Ok, tutti i dati inseriti nel form verranno salvati nella tabella relative alla visita compresa la chiave referenziale come è visibile nella funzione descritta di seguito :

```
NtDbConnection.Data.Table("tblVisite").AddRow(New Object()  
{mNewGuid, mPatientId, mTipologia,  
ctrlData.DateTimePicker1.Value.ToShortDateString, esaminatore})
```

Attraverso questo esempio è possibile notare come le viste create con una certa logica risultano veramente utili, in particolare la funzione già descritta CreateRow. Oltre a questa le altre funzioni principali utilizzate sono :

```
Public Function CreateTextboxColumn(ByVal readOnlyColumn As  
Boolean, ByVal columnName As String, ByVal columnHeader As String,  
ByVal columnSortMode As  
System.Windows.Forms.DataGridViewColumnSortMode, Optional ByVal  
columnTextAlignment As  
System.Windows.Forms.DataGridViewContentAlignment = MiddleLeft,  
Optional ByVal columnFillWeight As Integer = -1, Optional ByVal  
columnMinimunWidht As Integer = -1) As  
System.Windows.Forms.DataGridViewTextBoxColumn
```

Nell'analisi effettuata pre-sviluppo tra le problematiche prese in considerazione c'è stata quella relativa all'update dinamico delle viste. L'obiettivo è quello di aggiornarla ogni qualvolta vengano effettuate le operazioni di aggiungi modifica ed elimina, per fare questo è necessario ricomporla ogni volta interrogando la base di dati. Sfruttando le già presenti funzioni ne sono state implementate sul codice delle Sub più complesse per garantire il risultato prefissato nell'analisi:

```
RefreshDatagridPazienti(id)  
Private Sub RefreshDatagridAnagrafica()
```

```
Public Sub RefreshDatagridVisite(ByVal idPazient As String)
```

Facendo riferimento al form principale; queste due Sub vengono utilizzate rispettivamente per aggiornare dinamicamente la vista relativa all'anagrafica e la vista relativa alle visite dei singoli pazienti selezionati .

# -TESTING

Tipologia di testing:

Essendo il software sviluppato un prodotto di tipo borderline, e quindi non soggetta alla marcatura CE, è stato collaudato con la metodologia informale.

La metodologia prevede che il programmatore, completato lo sviluppo, e ad ogni modifica al software, manda in esecuzione il software e verifica interattivamente se il funzionamento è quello atteso. Se il comportamento è insoddisfacente, apporta altre modifiche e reitera il procedimento.

Quando il programmatore è soddisfatto del comportamento del software, invia il software ad altri colleghi, che effettuano un'ulteriore rapida prova. A questo punto, se non si tratta di modifiche che devono urgentemente essere rese operative, la nuova versione viene inviata agli utenti finali. Agli utenti finali viene illustrato il software con le modifiche apportate, questo tipo di approccio consente anche la rilevazione di ulteriori malfunzionamenti, oppure la richiesta di ulteriori modifiche. Infatti in tale procedimento informale di collaudo, la segnalazione di malfunzionamenti e di nuove versioni non segue un iter ben definito, di fatto si usano comunicazioni di persona, telefoniche, e-mail.

Il software seppure non ha avuto bisogno di metodologie di testing più rigide nei processi di validazione, ha comunque dovuto rispettare i requisiti di affidabilità e coerenza dei dati, perché gestendo dati anagrafici, è stato necessario creare a livello di sviluppo, una struttura solida, e a livello sistemistico creare copie di backup replicate.

Il video in allegato, mostra un esempio di aggiunta, modifica e cancellazione paziente

## .-CONCLUSIONI

Con questo progetto di tesi si è cercato di realizzare un software medicale, che rispecchi a pieno le funzionalità richieste dal cliente .

La maggior parte dei software che risiedono in particolare nelle strutture ospedaliere sono datati e sviluppati con vecchie tecnologie e spesso i dati non hanno una centralizzazione.

Questo progetto ha i suoi molteplici vantaggi sia in fase di sviluppo che nell'utilizzo da parte dell'utente.

La struttura dati seppur non sviluppata nel metodo convenzionale che prevede un'implementazione delle relazioni all'interno della stessa ,garantisce una maggior sicurezza stabilità elasticità e portabilità.

Le dll utilizzate hanno rivestito un ruolo molto importante perché :

- l'interazione con la banca dati risulta molto semplificata se paragonata alle classiche tecniche di programmazione e diminuiscono la percentuale di errori che spesso derivano dall'utilizzo della sintassi SQL

- le viste e i controlli permettono una miglior astrazione dei dati e un'interfaccia sicuramente più personalizzabile e user friendly per l'utente.

L'utilizzo di un server sql dedito e dei servizi di rete come l'active directory già presenti nella struttura ospedaliera permette la centralizzazione dei dati e l'opportunità di accedere ad essi da qualsiasi postazione attraverso le proprie credenziali sfruttando già quelle presenti, in modo da assicurare un accesso "controllato" e una banca dati sempre aggiornata e sincronizzata.

Questa progettazione potrebbe essere seguita passo per passo per poter ottenere un prodotto finito e commercializzabile.

Una volta che il software è definitivo ed è stato validato dal cliente si presume che bisognerà creare pacchetto di installazione (nell' appendice una guida per la creazione di un ipotetico pacchetto di installazione.)

# APPENDICE:

Nell'appendice vengono illustrate alcune porzioni di codice del programma relative alla gestione dei pazienti e la guida Intel su come creare un pacchetto installante

Codice relativo alla vista dell'anagrafica.

```
RefreshDatagridPazienti(id)
    Private Sub RefreshDatagridAnagrafica()

        dgvRegistryPatients.Datagrid.Rows.Clear()
        For Each row As DataRow In mAnagraficaTable.Rows
            Dim id As String = row.Item("ID")
            Dim nome As String = row.Item("Nome")
            Dim cognome As String = row.Item("Cognome")
            Dim sessoAsNumeric As Integer = row.Item("Sesso")

            Dim strSesso As String = ""
            Dim rowCollectionEsaminatore As DataRow() =
NtdbConnection.Data.Table("tblSETUPComboValues").SqlFilterFast("Co
mboName = 'Sesso'")
            For Each rowEsaminatore As DataRow In
rowCollectionEsaminatore
                If rowEsaminatore.Item("KeyNumericValue") =
sessoAsNumeric Then
                    strSesso = rowEsaminatore.Item("DisplayValue")
                End If
            Next

            Dim dataDiNascita As String =
row.Item("DataDiNascita")
            Dim luogoDiNascita As String =
row.Item("LuogoDiNascita")
            Dim codiceFiscale As String =
row.Item("CodiceFiscale")
            Dim indirizzoResidenza As String =
row.Item("ViaResidenza")
            Dim capResidenza As String = row.Item("CAPResidenza")
            Dim localitàResidenza As String =
row.Item("LocalitàResidenza")
            Dim provinciaResidenza As String =
row.Item("ProvinciaResidenza")
            Dim nazioneResidenza As String =
row.Item("NazioneResidenza")
```

```

        Dim scolarita As Integer = row.Item("Scolarità")
        Dim professione As String = row.Item("Professione")
        Dim telefono As String = row.Item("Telefono")
        Dim telefonoAlternativo As String =
row.Item("TelefonoAlternativo")
        Dim email As String = row.Item("Email")
        dgvRegistryPatients.CreateRow(New Object() {cognome,
nome, strSesso}, id)

        Next
End Sub

```

Codice relativo alla vista delle visite filtrate in base al paziente.

```

Public Sub RefreshDatagridVisite(ByVal idPazient As String)
    ' idPaziente = mPatientId.ToString
    Dim rowsVisite As DataView =
NtDbConnection.Data.SqlQuery("SELECT * FROM tbleMG_Archivio WHERE
tbleMG_Archivio.CODICE= '" & idPazient & "'ORDER BY
tbleMG_Archivio.ESEGUITO DESC")
    If rowsVisite.Count = 0 Then
        If MsgBox(" Non risultano diagnosi effettuate sul
paziente . Aggiungere una nuova diagnosi ?",
MsgBoxStyle.Information Or MsgBoxStyle.YesNo) = MsgBoxResult.No
Then
            Exit Sub
        Else
            mButtonVisiteMod = False
            Dim formEmgVisite As New
frmEMGvisita(mNameOfPatient, mPatientId, mVisitaId,
mButtonVisiteMod)
            formEmgVisite.ShowDialog()
        End If
    Else
        For Each row As DataRow In rowsVisite.Table.Rows
            Dim id As String = row.Item("ID")
            Dim nEsame As String = row.Item("N_ESAME")
            Dim data As String = row.Item("ESEGUITO")
            Dim provenienza As String = row.Item("PROVENIE")
            Dim sqlQueryVisite As String = "SELECT
tblSETUPComboValues.DisplayValue FROM tblSETUPComboValues WHERE
tblSETUPComboValues.ComboName = 'ProvenienzaEMG' AND
tblSETUPComboValues.KeyNumericValue = " & provenienza & ""

```

```

        Dim rowCollectionProvenienza As DataView =
NtdbConnection.Data.SqlQuery(sqlQueryVisite)
        provenienza =
rowCollectionProvenienza.Item(0).Item(0)
        dgvPatientEvaluations.CreateRow(New Object()
{nEsame, data, provenienza}, id)

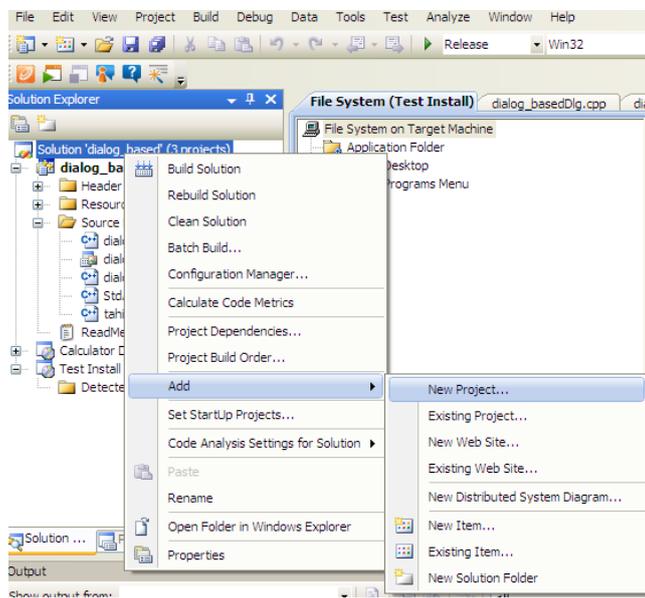
    Next
    ' Dim note As String = row.Item("Note")
End If
End Sub

```

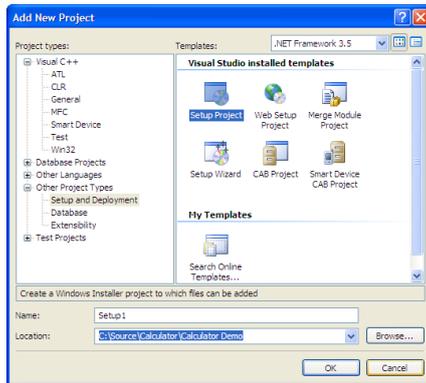
## -Guida Intel alla creazione del pacchetto installante

### Passaggio 1 Creare un progetto di installazione e distribuzione

Fare clic con il pulsante destro del mouse in Solution Explorer, a destra dell'IDE di Microsoft Visual Studio\*, quindi scegliere Add -> New Project. Nell'elenco di tipi di progetto scegliere Other Project Types -> Setup and Deployment -> Setup Project.

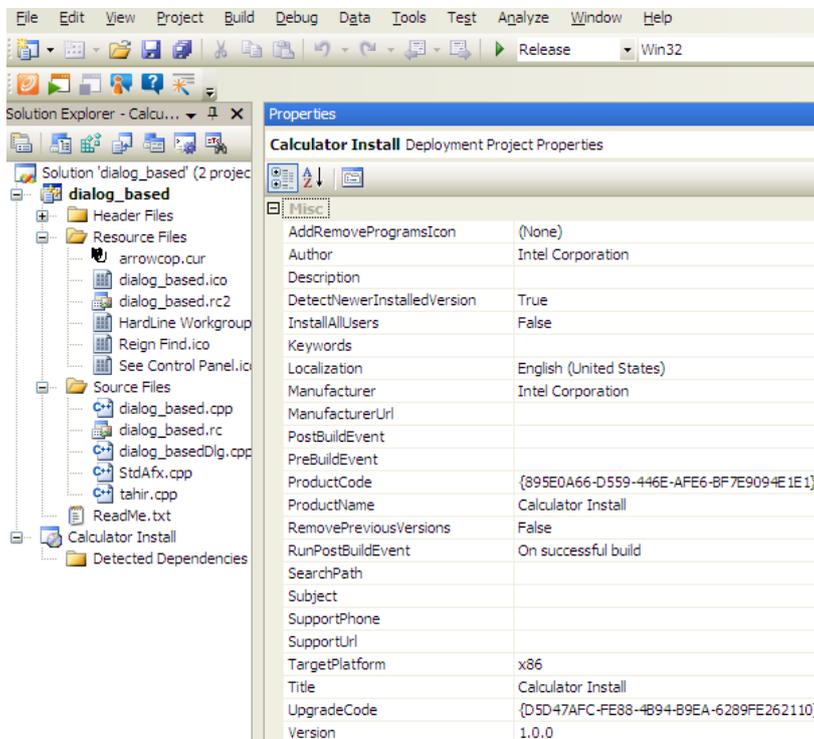


Aggiungere un nuovo progetto:



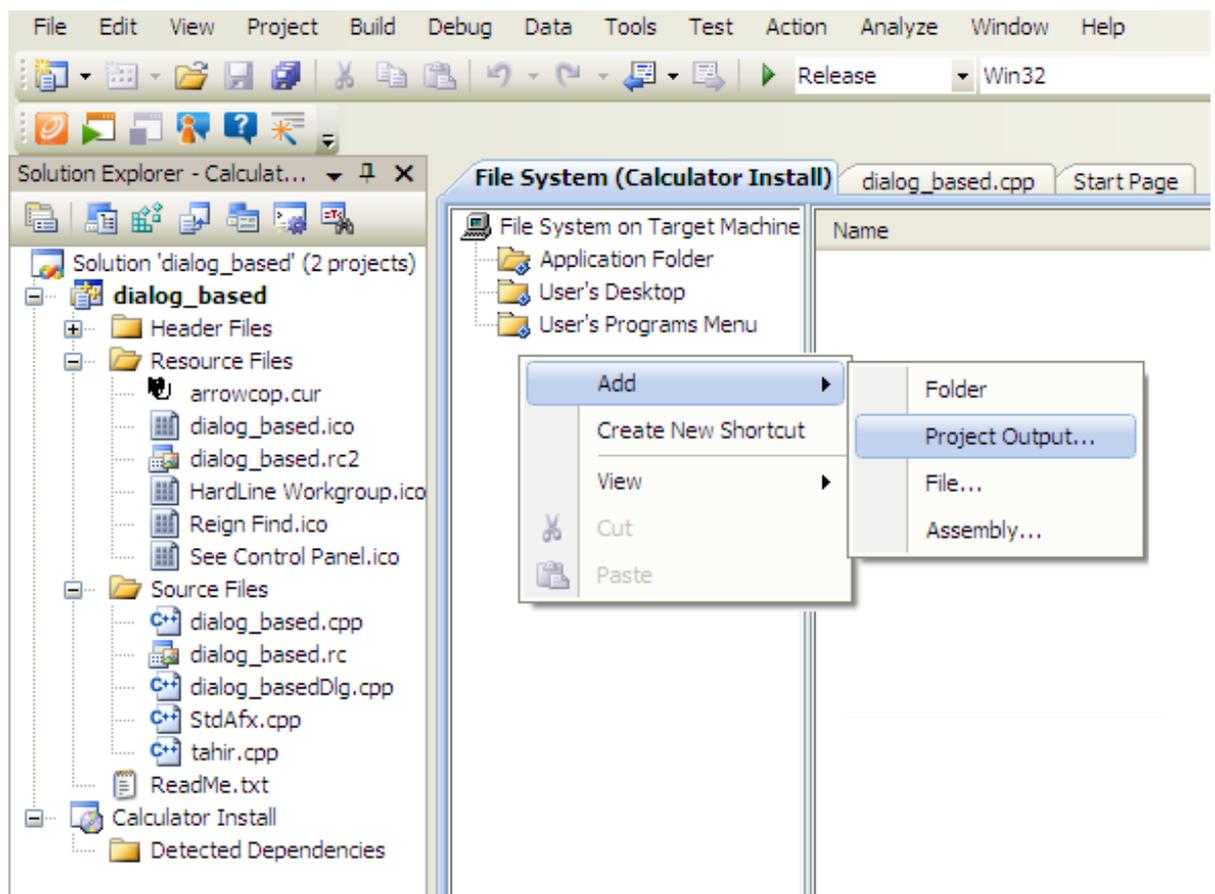
## Passaggio 2 Proprietà dell'installazione

A questo punto specifichiamo chi ha creato questo programma di installazione. Fare clic sul nuovo progetto di Installazione e distribuzione in Esplora soluzioni. Verrà visualizzata la finestra Proprietà (solitamente sotto Esplora soluzioni), con le impostazioni relative a Produttore, Autore e Titolo. Compilare questi campi con le informazioni appropriate.

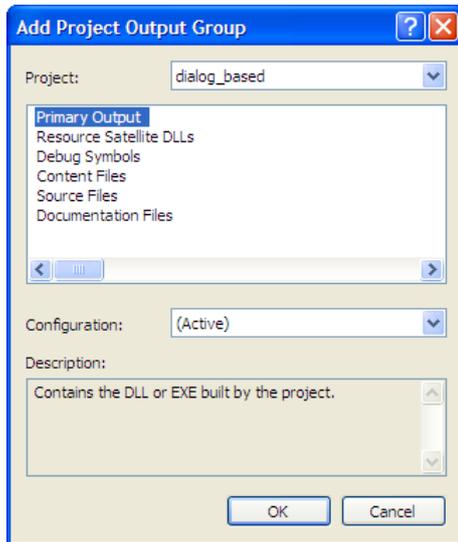


### Passaggio 3 Aggiunta di output e contenuto del progetto

Aggiungiamo nel progetto l'eseguibile (.exe) e tutti i file di contenuto richiesti. Fare clic con il pulsante destro del mouse sul progetto di Installazione e distribuzione in Esplora soluzioni e quindi scegliere View -> File system. Fare clic con il pulsante destro del mouse sul nodo Application Folder e quindi scegliere Add -> Project Output. Scegliere Primary Output, che includerà il file exe. Ripetere questo passaggio e scegliere Content Files se il progetto include contenuti richiesti quali immagini, audio, ecc.



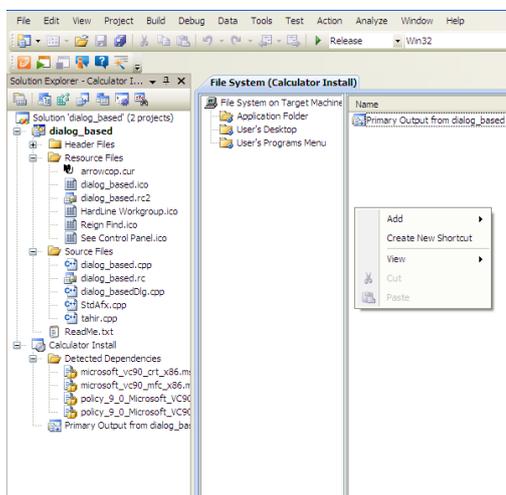
\*Nota speciale: nel progetto di applicazione, verificare che tutti i file richiesti siano contrassegnati come Content. A tale scopo, selezionare il file in Solution Explorer e impostare la proprietà Content su True.

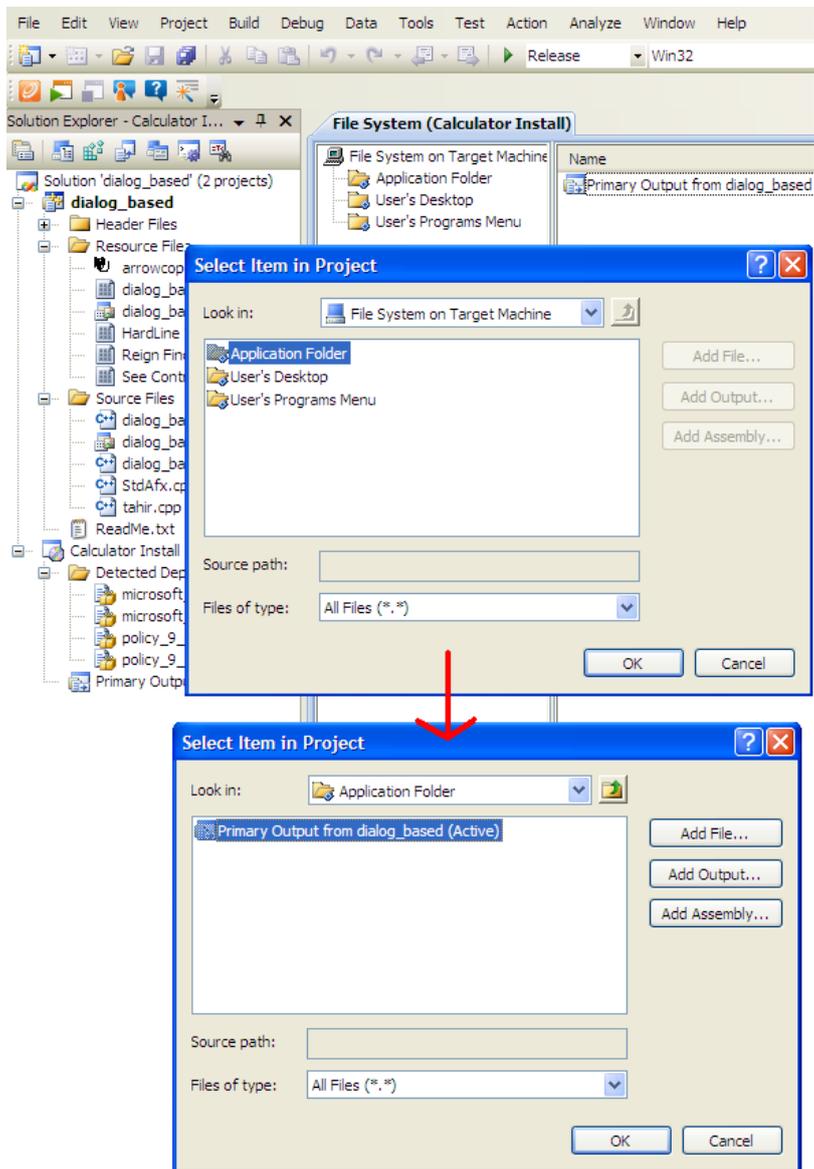


Dobbiamo includere l'icona da utilizzare in seguito nel collegamento. Fare clic con il pulsante destro del mouse sul nodo Application Folder e quindi scegliere Add -> File. Individuare il file dell'icona e fare clic su OK.

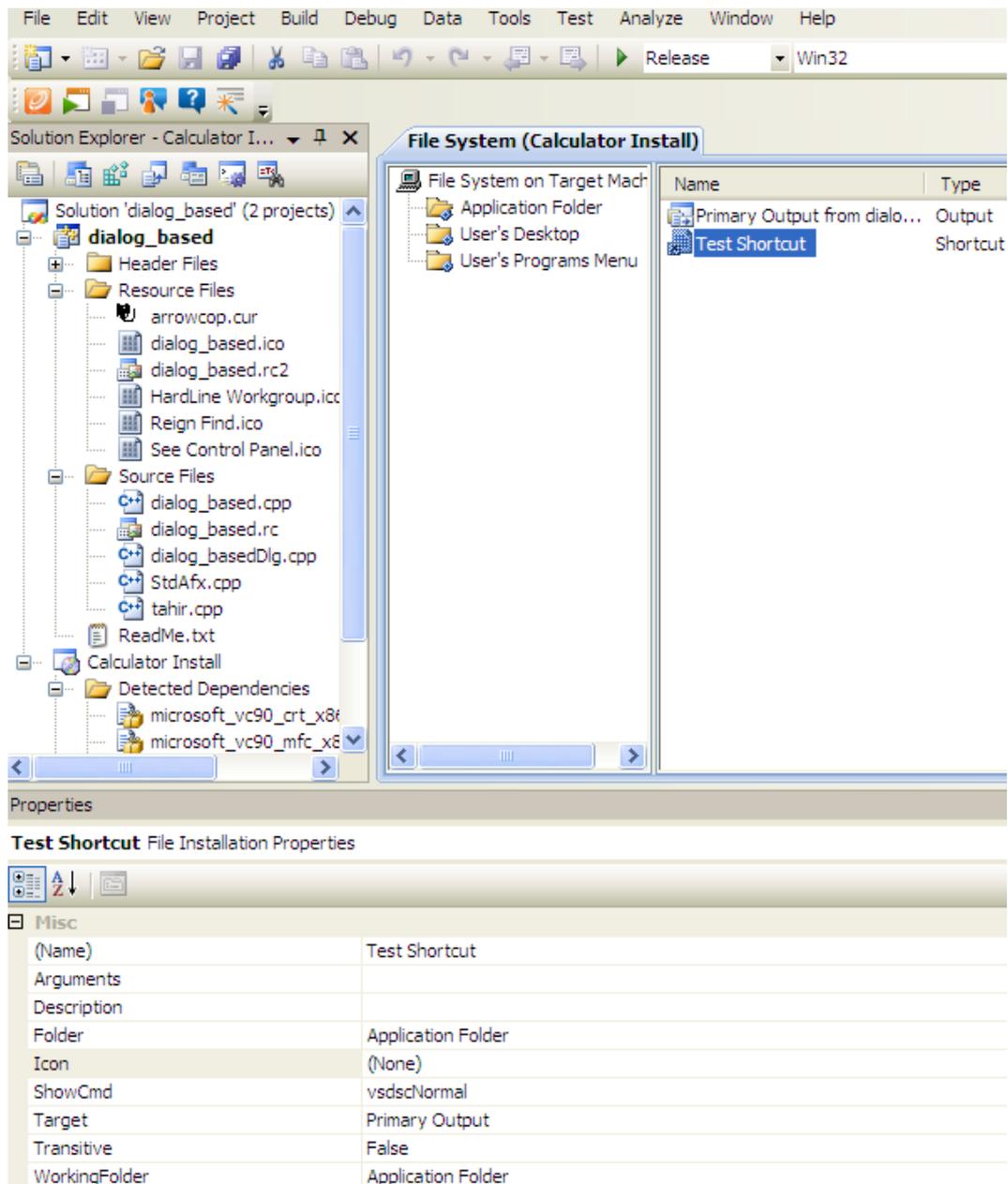
#### Passaggio 4 Creare i collegamenti

La presenza di collegamenti nell'applicazione migliora in modo significativo l'esperienza dell'utente. L'ideale è che l'applicazione sia facilmente accessibile da parte dell'utente finale. Lo si può realizzare creando vari collegamenti sul dispositivo. Fortunatamente, questa procedura è piuttosto semplice. Con la scheda File system di Installazione e distribuzione ancora aperta, fare clic con il pulsante destro del mouse nel riquadro destro per visualizzare un menu di scelta rapida. Scegliere Create New Shortcut. Da questa finestra di dialogo passare a Cartella applicazione e scegliere Primary Output da "Project name" (attivo). In questo modo verrà impostato il collegamento all'eseguibile. Fare clic su OK per accettare la selezione.





A questo punto è necessario specificare anche un'icona per il collegamento. Fare clic sul collegamento appena creato nel riquadro destro. Nella finestra Proprietà è possibile scegliere un'icona. Fare clic su Browse e passare alla Cartella applicazione, quindi selezionare l'icona aggiunta al file system nel precedente passaggio 3.



Se si desidera includere un collegamento nel menu Start, ripetere i passaggi precedenti sostituendo il nodo User's Programs Menu al nodo User's Desktop.

### Passaggio 6 Compilazione

Fare clic con il pulsante destro del mouse sul progetto di Installazione e distribuzione e quindi scegliere Build. Se la procedura è stata eseguita correttamente, sarà ora disponibile un file MSI appropriato nella cartella di output.

# -BIBLIOGRAFIA

[http://www.salute.gov.it/portale/temi/p2\\_6.jsp?lingua=italiano&id=2887&area=dispositivi-medici&menu=vuoto](http://www.salute.gov.it/portale/temi/p2_6.jsp?lingua=italiano&id=2887&area=dispositivi-medici&menu=vuoto)

[http://www.ssfa.it/allegati/radici\\_dm%20borderline\\_240211-01.pdf](http://www.ssfa.it/allegati/radici_dm%20borderline_240211-01.pdf)

<http://technet.microsoft.com/it-it/library/ms130214%28v=sql.100%29.aspx>

[http://www.mrwebmaster.it/sql-server/panoramica-sql-server-2008\\_10496.html](http://www.mrwebmaster.it/sql-server/panoramica-sql-server-2008_10496.html)

[http://it.wikipedia.org/wiki/Active\\_Directory](http://it.wikipedia.org/wiki/Active_Directory)

<http://technet.microsoft.com/en-us/library/bb742424.aspx>

[http://blog.pizzut.it/articolo\\_1073\\_Interagire-con-Microsoft-Word-tramite-Visual-Basic-Net-e-C-23-in-ambiente-Windows-Application-2E.html](http://blog.pizzut.it/articolo_1073_Interagire-con-Microsoft-Word-tramite-Visual-Basic-Net-e-C-23-in-ambiente-Windows-Application-2E.html)

<http://msdn.microsoft.com/it-it/library/52f3sw5c%28v=vs.90%29.aspx>

<https://software.intel.com/it-it/articles/how-creating-your-msi-installer-using-visual-studio-2008>

[http://it.wikipedia.org/wiki/Collaudo\\_del\\_software](http://it.wikipedia.org/wiki/Collaudo_del_software)