



IL SOFTWARE GESTIONALE

Tesina per Esame di Stato

Nicholas Boin

I.I.S. Primo Levi di Badia Polesine – Classe V Sez. A T.E.I.
Indirizzo Informatica e Telecomunicazioni – Articolazione Informatica
Anno Scolastico 2016 – 2017

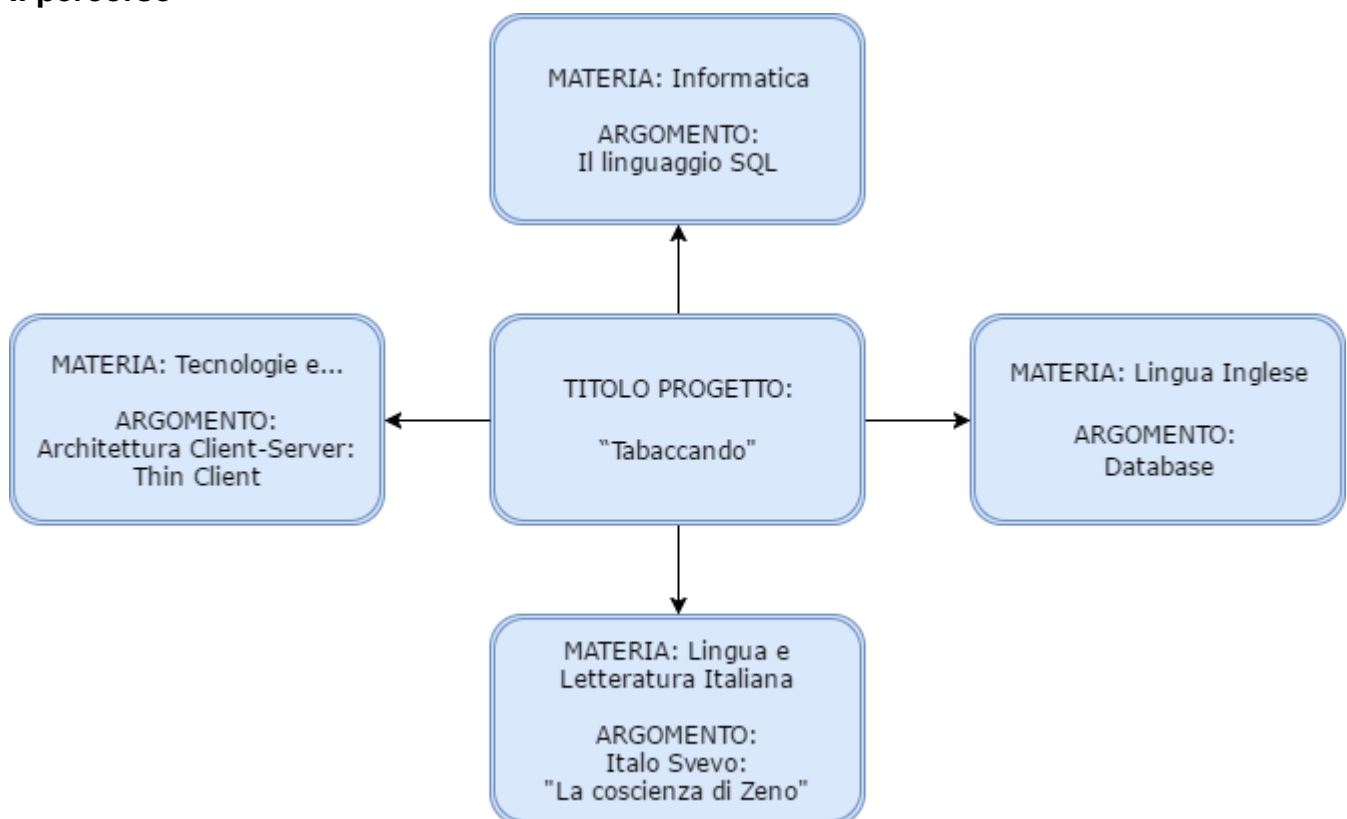
Introduzione.....	2
Il percorso.....	2
I collegamenti	2
IL PROGETTO – “Tabaccando”	3
Informatica	4
Il linguaggio SQL.....	4
Ddl – Data Definition Language	4
Dql – Data Query Language.....	5
Dml – Data Modification Language	6
Dcl – Data Control Language.....	6
Transazioni	6
Tecnologie e progettazione di sistemi informatici e di telecomunicazioni.....	7
Architettura Client-Server	7
Vantaggi del modello Client-Server	7
Svantaggi del modello Client-Server.....	7
Thin Client & Fat Client.....	8
Lingua Inglese	9
Databases	9
Relational Database Model	9
DBMS	9
Query	10
Lingua e Letteratura Italiana	11
Italo Svevo	11
La coscienza di Zeno	11
L’ultima sigaretta	13
Bibliografia.....	14

Il progetto “Tabaccando” è nato dall’esigenza di un software gestionale in grado di semplificare le procedure svolte dai tabaccai. La scarsa presenza in rete di software gestionali dedicati alle tabaccherie mi ha spinto a realizzare un software che, pur sembrando banale o già diffuso in forme più generali, è in grado di soddisfare le principali richieste nel settore delle tabaccherie.

Il gestionale, pur essendo mirato alla gestione del tabacco, integra funzioni aggiuntive più generali per la gestione delle vendite, della giacenza di magazzino, di pianificazione eventi e contabili.

Queste funzioni aggiuntive, assieme a quelle strettamente relative al tabacco, garantiscono ai tabaccai di trovare la maggior parte delle funzionalità che maggiormente utilizzano in un solo software invece che in software differenti, per i quali, spesso, si è reso necessario l’acquisto della licenza.

Il percorso



I collegamenti

Il software gestionale è composto essenzialmente di pagine web dinamiche e di un database, sono perciò stati collegati:

- il linguaggio SQL per la materia Informatica, in quanto parte fondamentale per l’interazione tra database e pagine web;
- il database per la materia Lingua Inglese, in quanto largamente utilizzato nel software per la memorizzazione di tutti i dati del gestionale;
- l’architettura Client-Server di tipo “thin client” per la materia Tecnologie e progettazione di sistemi informatici e di telecomunicazioni, in quanto il progetto è stato realizzato basandosi su tale architettura;
- “La coscienza di Zeno” di Italo Svevo per la materia Lingua e Letteratura Italiana, in quanto il progetto è stato realizzato per tabaccherie, le quali, per definizione, sono adibite alla vendita di tabacco, ovvero il punto debole più evidente del protagonista dell’opera “La coscienza di Zeno”.

“Tabaccando”, come già accennato, è un software gestionale per tabaccherie.

Con questo software un tabaccaio può:

- **visualizzare il listino dei tabacchi e ricercarli:** la sezione “Tabacco” è designata per la ricerca e visualizzazione dei dettagli dei tabacchi; è possibile ricercare i tabacchi tramite il codice AAMS, il quale contraddistingue univocamente tutti i tabacchi, oppure tramite descrizione; è inoltre possibile per ogni tabacco visualizzarne i dettagli.
- **visualizzare ed effettuare ordini di tabacco:** la sezione “Ordine” permette di creare ordini di tabacco, eventualmente modificarli od eliminarli (se non ancora spediti) e visualizzare l’elenco degli ordini effettuati dalla tabaccheria ed il dettaglio di ogni singolo ordine.
- **gestire la propria giacenza nel magazzino:** la sezione “Magazzino” permette di registrare i prodotti presenti nel magazzino in modo da permettere al tabaccaio di monitorare le quantità di prodotti rimaste ed accorgersi di eventuali prodotti scarsi o non presenti oltre a quelli già presenti; è inoltre possibile settare il software in modo che l’ordine di tabacco evaso venga aggiunto automaticamente al magazzino semplificando le operazioni di inserimento del tabaccaio.
- **registrare le vendite dei tabacchi:** la sezione “Vendite” permette di registrare le vendite dei tabacchi scalandoli automaticamente dalle presenze in magazzino, aiutando a ricordare quali prodotti scarseggiano e nel calcolo contabile;
- **gestire la contabilità:** la sezione “Statistiche” permette di visualizzare la quantità di tabacchi venduti, mostrando le statistiche di vendita per prodotto e gli incassi ottenuti.
- **effettuare calcoli con semplicità:** la sezione “Calcolatrice” permette di effettuare calcoli senza dover lasciare il gestionale; sono presenti, inoltre, le “funzioni” più utilizzate dai tabaccai, come l’IVA.

Il gestionale si presenta sotto forma di una **piattaforma per vari tabaccai** invece che come software dedicato ad una singola tabaccheria; è infatti possibile registrare più tabaccherie al software definendo per ciascuna di queste ulteriori utenti aventi accesso esclusivamente alla propria tabaccheria. Ogni tabaccheria sarà gestita da un solo utente configurato come *amministratore* per tale tabaccheria; questo utente potrà poi gestire gli altri utenti della tabaccheria oltre alla tabaccheria stessa.

Il sistema software non implementa attualmente un metodo di **controllo di validità** delle tabaccherie inserite nel sistema in quanto un valido metodo per garantire la corretta registrazione delle tabaccherie inserite consisterebbe nell’invio, all’indirizzo specificato via posta, un codice di verifica da inserire nel sistema per validare la registrazione, dopo aver controllato, tramite un accordo con l’Agenzia dei Monopoli, la reale e fiscale esistenza dell’esercizio commerciale. Da quanto sopra riportato, è possibile considerare una reale implementazione di tale meccanismo in caso di reale ed effettiva intenzione di realizzazione del software in questione.

Citando nuovamente gli eventuali accordi con l’Agenzia dei Monopoli, è importante sottolineare che nella versione del software sviluppata attualmente il database dei tabacchi è **inserito a mano dal gestore del sito**; un eventuale accordo con questa Agenzia potrebbe automatizzare, invece, l’inserimento e l’aggiornamento del database dei tabacchi, accordo difficilmente realizzabile attualmente.

Un altro miglioramento per il software potrebbe essere costituito dall’utilizzo dei **codici a barre** dei pacchetti di sigarette per la vendita del tabacco, risultando più rapido l’inserimento delle vendite da parte del tabaccaio; tuttavia per realizzare ciò si renderebbe necessario l’acquisto di uno scanner per codice a barre esterno, spesa attualmente non prevista data l’attuale significatività del progetto.

Questo software gestionale si presenta **attualmente come portale gratuito** per le tabaccherie; tuttavia, è da considerare l’idea di una remunerazione per gli sviluppatori in caso di effettiva realizzazione ed ufficializzazione del progetto, attraverso il pagamento di una somma di denaro accessibile in un’unica soluzione alla registrazione di una tabaccheria, ammontante ad esempio a €100.

IL LINGUAGGIO SQL

Il linguaggio SQL fu inventato nel 1974 dall'IBM, inizialmente con il nome "Sequel", per lavorare con i database relazionali; nella versione successiva venne detto "SQL", diventando il linguaggio standard per l'interazione con i database relazionali. Sebbene "SQL" sia l'acronimo di Structured Query Language, il linguaggio SQL non permette solamente di effettuare query; questo linguaggio, infatti, consente:

- DDL, Data Definition Language, permette di definire schemi relazionali, eliminare relazioni e modificare schemi relazionali;
- DML, Data Manipulation Language, permette di inserire, modificare od eliminare record in relazioni;
- DQL, Data Query Language, permette di interrogare il database e di operare sui risultati dell'interrogazione;
- DCL, Data Control Language, permette di creare e gestire gli utenti e i relativi permessi per le funzioni precedentemente citate di operazioni su database.

Le istruzioni SQL vengono tutte denominate, in modo informale, "query" anche se non tutte sono effettivamente query.

DDL – DATA DEFINITION LANGUAGE

Le istruzioni SQL DDL permettono di creare o modificare la struttura degli schemi relazionali o di eliminare relazioni o database.

Per definire un nuovo database si usa:

```
CREATE DATABASE IF NOT EXISTS nome_database ;
```

La clausola "IF NOT EXISTS" specifica l'esecuzione dell'istruzione solo nel caso in cui il database di nome specificato non esista già.

Per definire una nuova relazione si usa:

```
CREATE TABLE IF NOT EXISTS nome_tabella ( nome_attributo_1 tipo vincoli_aggiuntivi,
      nome_attributo_2 tipo vincoli_aggiuntivi, ... ,
      PRIMARY KEY (nome_attributo) ,
      FOREIGN KEY (nome_attributo) REFERENCES relazione_esterna (nome_attributo_relaz_est) ) ;
```

Questa istruzione permette di creare una nuova tabella con nome specificato e contenente gli attributi elencati tra le parentesi aventi ognuno uno specifico tipo ed eventuali vincoli aggiuntivi; ad esempio i ricorrenti "NOT NULL" o "AUTO_INCREMENT" (per il tipo intero). "PRIMARY KEY" specifica l'attributo o gli attributi da utilizzare come chiave primaria, mentre "FOREIGN KEY" specifica un attributo da utilizzare come chiave esterna con l'opportuno riferimento all'attributo della relazione esterna.

I tipi standard di base supportati da SQL sono:

- INT: numero intero;
- SMALLINT: numero intero piccolo (il valore massimo dipende dalla macchina di esecuzione)
- FLOAT(*n*): numero in virgola mobile con una precisione almeno *n* cifre;
- NUMERIC(*p*, *d*): numero con precisione specificata dall'utente, dove *p* indica il numero delle cifre e *d* il numero di cifre di *p* alla destra della virgola;
- CHAR(*n*): stringa di lunghezza fissa di *n* caratteri;
- VARCHAR(*n*): stringa di lunghezza variabile di massimo *n* caratteri.

Per eliminare un database si usa:

```
DROP DATABASE nome_database ;
```

mentre per eliminare una relazione si utilizza:

```
DROP TABLE nome_tabella ;
```

Per modificare lo schema relazionale di una tabella esistente è possibile eliminare od aggiungere attributi, con la conseguente perdita dell'attributo in tutti i record in caso di eliminazione; in caso di aggiunta viene inserito l'attributo con il valore di default `NULL` in tutti i record. Le istruzioni per aggiungere ed eliminare attributi sono:

```
ALTER TABLE nome_tabella ADD nome_attributo ;  
ALTER TABLE nome_tabella DROP nome_attributo ;
```

DQL – DATA QUERY LANGUAGE

Le istruzioni SQL DQL permettono l'interrogazione dei dati nel database.

L'istruzione per la query di base è così strutturata:

```
SELECT attributo1 AS nome_attributo1, attributo2 AS nome_attributo2, ...  
FROM tabella1 nome_tabella1, tabella2 nome_tabella2, ...  
WHERE condizione ;
```

In questa query è possibile effettuare la proiezione delle colonne da visualizzare dopo il "SELECT" (con "*" si indicano tutte le colonne), specificando i database di provenienza dopo il "FROM" e dichiarando la condizione di selezione dopo il "WHERE". In caso di utilizzo di più tabelle, queste, solitamente, vengono rinominate dopo il "FROM" per poter essere indicate facilmente davanti agli attributi separate da un punto. Dopo il "SELECT" è possibile specificare anche un metodo:

- "ALL": specifica di selezionare tutti i record, anche duplicati;
- "DISTINCT": specifica di eliminare i record duplicati;
- "TOP": specifica di selezionare solo il primo record.

È possibile anche specificare funzioni di aggregazione di base nel "SELECT" come "SUM" per la somma dei valori nella colonna proiettata, "AVG" per la media, "MIN" per il valore minimo, "MAX" per il valore massimo, "COUNT" per il numero di record selezionati.

La condizione posta dopo il "WHERE" è una condizione booleana, ad esempio: "*nome_attributo=valore_input_utente*" oppure "*nome_attributo* IS NOT NULL"; in questo modo verranno selezionati solamente i record che sono in grado di soddisfare la condizione posta. È possibile utilizzare i connettivi logici per porre altre condizioni, come "AND", "OR"...

Particolari costrutti permettono inoltre di lavorare con le stringhe: la clausola "LIKE" permette di comparare dei pattern per individuare le informazioni richieste, consentendo l'utilizzo al suo interno di caratteri speciali per confrontare ed individuare gli attributi con la stringa che soddisfa il pattern; il carattere speciale "%" indica una qualsiasi sottostringa di qualsiasi lunghezza, anche 0, invece il carattere "_" indica un singolo carattere qualsiasi, in ogni caso di lunghezza 1 e non 0. Si ricordi che il backspace "\" permette di rendere caratteri normali i caratteri speciali precedentemente indicati.

Il "NATURAL JOIN" è un comando particolare che è possibile costruire anche seguendo la struttura della query appena analizzata, ad esempio:

```
SELECT attributo1 AS nome_attributo1, attributo2 AS nome_attributo2, ...  
FROM tabella1 tab1, tabella2 tab2, ...  
WHERE tab1.chiave_esterna = tab2.chiave_primaria AND condizione_ulteriore ;
```

Oltre alle clausole finora analizzate, è possibile inserire altre clausole alla fine:

- "GROUP BY": funzione di aggregazione che permette di raggruppare insieme i record in base ad un attributo;
- "HAVING": condizione da imporre sul "GROUP BY", equivalente ad un "WHERE", però, effettuato sul raggruppamento;
- "ORDER BY": specifica di ordinare in modo crescente i risultati della query secondo l'attributo specificato; per ordinare in maniera decrescente è sufficiente aggiungere "DESC" alla fine;

- “LIMIT *n*”: specifica di selezionare solo i primi *n* record selezionati dalla query.

È inoltre possibile creare sotto-query, in modo da utilizzare i risultati di una sotto-query in una query esterna; di seguito è fornito un esempio:

```
SELECT attributo_tabella, attributo2_tabella
FROM tabella
WHERE attributo_tabella IN ( SELECT attributo
                             FROM tabella
                             WHERE condizione ) ;
```

DML – DATA MODIFICATION LANGUAGE

Le istruzioni SQL DML permettono di creare, modificare o eliminare record da una tabella in un database.

Per l’inserimento di record si utilizza:

```
INSERT INTO nome_tabella (attributo1, attributo2, attributo3)
VALUES ("valore1_attributo1", "valore1_attributo2", "valore1_attributo3"),
("valore2_attributo1", "valore2_attributo2", "valore2_attributo3") ... ;
```

Nell’istruzione appena descritta è possibile omettere l’elenco delle colonne solo nel caso in cui i valori inseriti siano quelli di tutte le colonne e siano nell’ordine corretto come indicato dallo schema relazionale della tabella interessata.

Per la modifica si utilizza invece:

```
UPDATE nome_tabella SET attributo1="valore_attributo2" , attributo2="valore_attributo2", ... ;
```

Per la cancellazione si utilizza infine:

```
DELETE FROM nome_tabella WHERE condizione;
```

In questo caso vengono cancellati tutti i record che soddisfano la condizione specificata.

DCL – DATA CONTROL LANGUAGE

Le istruzioni SQL DCL permettono di controllare i permessi assegnati agli utenti per operare su determinati database.

Le istruzioni utilizzate per assegnare o revocare i permessi sono:

```
GRANT privilegio1, privilegio2, ... ON nome_database TO utente;
REVOKE privilegio1, privilegio2, ... ON nome_database FROM utente;
```

In queste istruzioni i privilegi assegnabili/revocabili sono corrispondenti ai comandi SQL che operano sui database, ad esempio SELECT, INSERT, UPDATE, DELETE, REFERENCES, ALTER, CREATE; specificando invece ALL si indicano tutti i permessi disponibili, creando una sorta di “amministratore” del database o revocando tutti i permessi per il database specificato.

TRANSAZIONI

Infine, è possibile fare utilizzo di transazioni, ovvero operazioni su database da eseguirsi in modo atomico o, in caso di problemi, annullare tali operazioni tornando allo stato precedente. I comandi caratteristici delle transazioni sono:

```
BEGIN TRANSACTION nome_transazione;
COMMIT [TRANSACTION] nome_transazione;
ROLLBACK [TRANSACTION] nome_transazione;
```

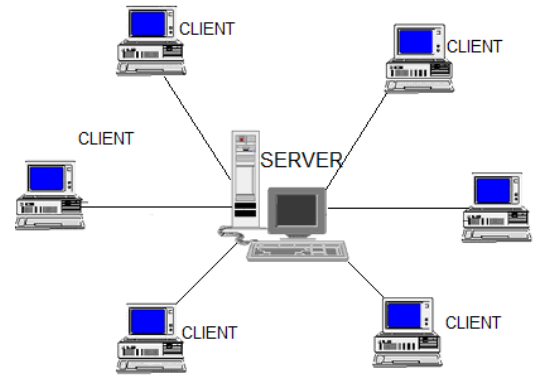
In questo caso il comando “BEGIN TRANSACTION” inizia una transazione, il comando “COMMIT” la inserisce nel database confermandola, mentre il comando “ROLLBACK” la annulla tornando allo stato precedente all’inizio della stessa.

ARCHITETTURA CLIENT-SERVER

Il modello Client-Server costituisce una delle architetture di rete più utilizzate nel World Wide Web.

Questa architettura è costituita principalmente da due tipi di host: il server ed il client.

- Il **server** è un particolare host o software utilizzato per mettere a disposizione uno o una serie di servizi, in base alla configurazione decisa dal sistemista;
- Il **client** è un host o software in grado di connettersi ad un server per usufruire dei servizi messi a disposizione.



Tipica rappresentazione di un'architettura Client-Server con topologia a stella

Generalmente il server con accezione hardware è una macchina ad elevate qualità, affidabilità e prestazioni in grado di sostenere connessioni multiple da più client, i quali dovendo solamente utilizzare il servizio di un server, sempre con l'accezione di hardware, sono macchine meno complesse e con prestazioni inferiori dei server. Facendo riferimento invece all'accezione software, un server è un processo in grado di fornire servizi ad altri processi client, non necessariamente nella stessa macchina fisica. In base al tipo di servizi vengono adottati specifici protocolli, ad esempio:

- protocollo HTTP per i Web Server (utilizzati per le pagine Web);
- protocolli SMTP, IMAP, POP3 per i Mail Server;
- protocolli FTP, SMB per i File Server;
- protocollo DNS per le risoluzioni di domini.

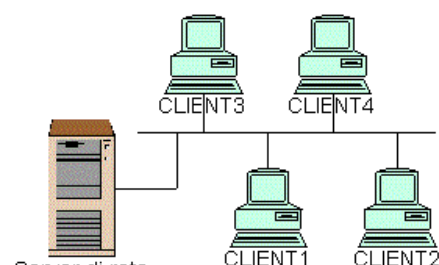
I protocolli sopra citati sono quelli standard e più comuni; è infatti possibile che, ad esempio, all'interno di un'azienda possano essere utilizzati protocolli personalizzati e proprietari sfruttando l'architettura Client-Server. Oggigiorno il modello server-client è applicato soprattutto per i servizi Web.

VANTAGGI DEL MODELLO CLIENT-SERVER

- Centralizzazione delle risorse: il server, concettualmente al centro della rete, permette l'accesso condiviso alle risorse che è in grado di conservare e può permettere a più client di accedere alla medesima risorsa, secondo le regole di condivisione ed accesso in precedenza definite;
- Indipendenza dai client: in caso di malfunzionamento di qualche client il server continua a fornire i suoi servizi senza problemi;
- Sicurezza: è possibile tramite appositi sistemi di sicurezza (ad esempio firewall, ACL, permessi ed altro) limitare l'accesso e/o la modifica alle risorse del server;
- Semplicità di configurazione: è necessario configurare solo il server che eroga il servizio, i client poi rispetteranno le regole impostate nel server;
- Manutenibilità: è sufficiente effettuare le operazioni di manutenzione solo sul server affinché possa essere affidabile per l'erogazione dei servizi desiderati;
- Scalabilità: l'aggiunta di client non arrecherà danni o malfunzionamenti alla rete.

SVANTAGGI DEL MODELLO CLIENT-SERVER

- Scarsa tolleranza ai guasti: in caso di guasti, se non sono presenti contromisure adatte a garantire (ad esempio, gruppi di continuità, sistemi RAID...) la continuità di funzionamento del server, la caduta del server causa il blocco del funzionamento dell'intera rete;
- Nel caso in cui la topologia della rete ponesse una sola "route" per raggiungere il server e questa si interrompesse, il server risulterebbe irraggiungibile dai client compromettendo il funzionamento corretto della rete (vedi figura a lato);



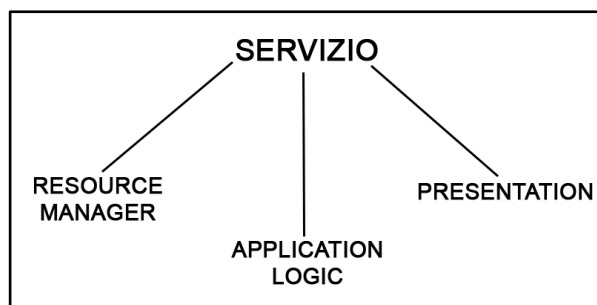
Architettura Client-Server in una rete con topologia a bus

- Costi: la necessità di una macchina prestante dedicata all'erogazione di servizi a tutti i client rappresenta un costo più elevato.

THIN CLIENT & FAT CLIENT

I servizi offerti da un server si compongono fondamentalmente di tre parti:

- **resource manager**: parte che gestisce le risorse e la loro memorizzazione fisica;
- **application logic**: parte per la gestione delle richieste, ricerca ed elaborazione dei dati;
- **presentation**: parte che si occupa della visualizzazione all'utente dei dati e risorse.



In base alle funzioni svolte dal client, si possono classificare:

- **thin client**: client che si occupano solo della funzione “presentation”, lasciando il “resource manager” ed “application logic” al server;
- **fat client**: client che si occupano anche di una parte di “application logic” oltre alla funzione “presentation”, lasciando la parte restante di “application logic” e “resource manager” al server.

Inizialmente i client erano di tipo “thin”, ma, con l’avanzare del progresso tecnologico, i client hanno acquisito potenze di calcolo maggiori tanto da poter supportare senza problemi una configurazione di tipo “fat”.

I **vantaggi** del “**thin client**” sono:

- compatibilità con client poco prestanti;
- semplicità di implementazione: il client deve solo mostrare i dati risultanti ricevuti;
- migliori performance per calcoli pesanti: essendo il server una macchina dalle alte performance è in grado di compiere pesanti elaborazioni di dati più velocemente di quanto potrebbe elaborare un client in configurazione “fat”.

I **vantaggi** del “**fat client**” invece sono:

- minore congestione di rete: essendo i dati elaborati dal client, vengono inviati solo i dati strettamente necessari per l’elaborazione invece che il risultato intero;
- minor carico di lavoro per il server: elaborando parte dei dati il client, il server non necessita di elaborare completamente i dati potendo mettere la potenza di calcolo risparmiata a disposizione di altri client;
- funzionamento offline: avendo una parte di elaborazione nel client, alcune operazioni possono essere svolte senza richiedere l’uso del server;
- migliori performance per file multimediali;
- retro-compatibilità con il “thin client”: avendo capacità di calcolo maggiori del “thin client”, quindi non inferiori, il “fat client” può funzionare anche in configurazione “thin”.

My project, called “Tabaccando”, is a type of management software for tobacconists, which means that an important part is performed by the database in order to store the information about tobacco, the tobacconists’ orders and eventually all the tobacco or other products sales. After storing this information the management software can perform calculations and operate with that data.

Focusing on the data storing aspect, this software uses a popular DBMS called “MySQL” to store information on a relational database.

The basic concepts involved in the project are explained below.

DATABASES

A database is an organised collection of information that can easily be accessed, managed and updated.

There are different kinds of database models which define the data organisation:

- Hierarchical model: with a one-way tree-like structure, it’s now old and outdated;
- Network model: with a two-way tree-like structure, it’s more efficient than the hierarchical model but obsolete;
- Relational model: composed by tables with rows and columns, it’s now the mostly used model;
- Object-oriented model: an improved model that organises data in an object oriented way.

It’s important to say that the database refers to the data and not to the manager software which manipulates it.



RELATIONAL DATABASE MODEL

The relational model was invented by Edgar Codd in 1970s. The relational database is a set of tables, called “relations”, each one composed of columns and rows; the columns are the “attributes” (or “fields”) and the rows are the “records” (or “instances”).

Attributes are specific types of information, such as numbers, strings of text, dates, times and so on. A record contains all the entries representing the properties of a table element. This model needs the definition of one or more columns as “primary” or “composite” keys which permit the entries to be uniquely identified in the entire table. In this model it’s also possible to define a column as “foreign key” where entries under this column match the primary or composite key of another table; this link between tables is usually called “association” or “relationship”.

The following rules define important integrity constraints useful to grant data consistency and reliability. For example:

- defining attribute types (attribute domains) reduces the risk of wrong data insertion;
- using a primary or composite key reduces redundancy so granting the uniqueness of a record;
- rejecting null values in primary keys ensures entity integrity;
- allowing only a foreign key that matches a really existing primary/composite key of another relation grants referential integrity.

Thanks to these rules the relational model is considered the most reliable and practical to implement, so it is now the most commonly used database model.

DBMS

A DataBase Management System is a kind of software used to create, manage and access a database.

A DBMS must ensure:

- The persistence of data: data is stored in a database on a mass storage device and survives after the closing of a program;
- Data availability: data must be available regardless of the existence of other concurrent users who are accessing the database;

- Data sharing: multiple users should be able to work on the same data simultaneously;
- Reliability: data must be consistent, that means data must have values that are really admissible and acceptable;
- Resistance to failures: data must be preserved as intact as possible even in case of system errors or hardware failure damages; if data loss occurs the DBMS must restore the last valid backup copy;
- Data handling: the DBMS must provide a method to access and manipulate data from databases through the use of a specific query language.
- Data independence: data is defined independently from the applications that interact with it, so any modification to the data structure doesn't affect the applications and vice-versa;
- Data integrity: the DBMS must provide mechanisms to check and validate submitted data in order to respect the specified integrity rules.

The DBMS works as intermediary software between the database and the applications which have to operate with it. The DBMS also manages the simultaneous access of multiple users to the same database.

The operations with the database are physically performed by the DBMS; these operations are specified by the application program through the "queries".

Some examples of DBMS are Oracle, Microsoft SQL Server, MySQL, PostgreSQL and Microsoft Access.

MySQL is the most famous RDBMS (Relational DBMS) with no user-friendly graphic interface, while Microsoft Access is the most commonly used DBMS with a user-friendly interface.

QUERY

A query is an instruction used to retrieve data from a database; technically speaking, queries perform only data retrieving but they are commonly used also for:

- defining or deleting databases (DDL) ;
- defining, modifying or deleting table structure (DDL) ;
- inserting, updating or deleting data from the database (DML) ;
- specifying users' privileges (DCL) .

The programmer must specify in which columns he wants to operate and the conditions for the selection of the entries.

The most commonly used query language is called "SQL" (Structured Query Language).



ITALO SVEVO

Italo Svevo, pseudonimo di Aron Hector Schmitz, nasce nel 1861 a Trieste, allora città "asburgica", da padre tedesco (Franz) e madre italiana (Allegra Moravia). Compie i suoi studi in Germania, in Baviera, dove studia tedesco e materie propedeutiche agli studi commerciali conclusi poi a Trieste. Conseguito il diploma, trova subito occupazione in una banca; nel frattempo, scrive e pubblica a sue spese due romanzi: **"Una vita"** nel 1892 e **"Senilità"** nel 1898, che, inizialmente, non riscuotono successo. Nel 1896 si sposa con Lidia Veneziani ed inizia a lavorare nella fabbrica di vernici del suocero. Dal matrimonio nasce la figlia Letizia. Agli inizi del '900 si iscrive ad una scuola di Inglese a Trieste dove conosce J. Joyce, prima suo insegnante, poi grande amico. Nel 1923 pubblica il suo terzo romanzo **"La coscienza di Zeno"** riscuotendo un notevole successo grazie anche all'appoggio di Joyce e alla recensione positiva di E. Montale. Muore in un incidente d'auto nel 1928 a Motta di Livenza lasciando incompiuto il romanzo, *continuazione* del precedente, dal titolo **"Il vecchione o Le confessioni del vegliardo"**. Un'opera postuma di grande interesse è l'**epistolario** dove emerge la concezione della vita dell'autore.



I. Svevo è considerato dalla critica un **autore mitteleuropeo**, cioè uno scrittore che si fa portavoce di più espressioni culturali europee; ad esempio, introduce ed approfondisce la tematica della **psicanalisi** dopo la traduzione del testo "Interpretazione dei sogni" di Sigmund Freud.

LA COSCIENZA DI ZENO

"La coscienza di Zeno" è il terzo romanzo scritto da Italo Svevo dal 1919 al 1922 e pubblicato nel 1923.

Il romanzo si compone in totale di otto capitoli, i quali, non considerando la *Premessa* e la *Prefazione*, non seguono un ordine cronologico ma costituiscono "episodi" tratti dalla vita del protagonista, Zeno Cosini.

Capitolo I. Prefazione

In questo capitolo viene introdotta la storia con il dottor S., analista del paziente Zeno Cosini, il quale afferma di pubblicare gli appunti del paziente per vendetta dopo che costui ha abbandonato le sue terapie ferendolo nel suo orgoglio professionale; gli appunti contengono gli episodi di vita del paziente scritti di suo pugno su consiglio del dottor S. per il suo percorso di cura dalla malattia psichica di cui era affetto.

Capitolo II. Preambolo

Zeno Cosini, come suggerito dal *dottore*, cerca di scrivere i propri ricordi di vita; dopo poco tempo, però, si addormenta e ricorda i suoi primi momenti. Inizialmente, nemmeno si riconosce; poi pensa a quante combinazioni di eventi possono causare dolore e preoccupazioni a un bambino privandolo dei giorni puri. Zeno, non avendo ottenuto il risultato desiderato, decide di ritentare il giorno seguente.

Capitolo III. Il fumo

In questo capitolo Zeno narra del vizio che lo tormenta da quando era ragazzino e dal quale non è mai riuscito a liberarsi: il fumo. Per cercare di liberarsi dal vizio del fumo, Zeno si fa ricoverare anche una clinica dalla quale, però, fugge corrompendo con una bottiglia di cognac l'infermiera che lo sorveglia.

Zeno analizza in questo capitolo la sua incapacità e mancanza di determinazione nel cercare di perseguire un fine con successo; questa sua incapacità, o meglio *inettitudine*, viene giustificata dal suo senso di vuoto interiore e dalla mancanza di una figura paterna determinata e rigorosa nella sua infanzia.



Capitolo IV. La morte di mio padre

In questo capitolo Zeno narra gli ultimi momenti del padre facendo riferimento anche ai rapporti che aveva con lui. Il protagonista cerca di accudire il padre tentando anche di stabilire con lui un rapporto affettivo prima che sia troppo tardi; tuttavia, una notte, il padre viene colpito da un edema cerebrale che lo rende incapace di intendere e volere, destinandolo a morte certa, prima di parlare seriamente con il figlio. Proprio prima di morire il padre schiaffeggia addirittura il figlio; questo gesto segna profondamente Zeno incapace di capirne il perché.

Capitolo V. La storia del mio matrimonio

In questo capitolo Zeno narra le vicende che lo hanno portato al matrimonio. Il protagonista, infatti, aveva cominciato a frequentare la famiglia di Giovanni Malfenti, uomo d'affari triestino, con la speranza di sposare la figlia più graziosa, Ada, che, però, lo rifiutò per la sua incapacità di cambiare. Sentendo il bisogno di una donna al suo fianco, fece la proposta di matrimonio anche alle altre sorelle: Anna non fu considerata perché di soli otto anni, Alberta lo respinse a sua volta mentre Augusta, la meno attraente, accolse la sua proposta rivelandosi poi la giusta moglie per Zeno. Il protagonista scrive di aver trovato in Augusta la figura dolce, materna e di conforto del quale aveva bisogno; ciò, comunque, non gli impedì di avere un'amante, Carla Greco.

Capitolo VI. La moglie e l'amante

La relazione con l'amante Carla, pur essendo diventata passionale da un inizio di attrazione prettamente fisica, si presenta ambigua per Zeno in quanto non vuole rinunciarvi ma non vuole nemmeno ferire la moglie amata. La storia con l'amante si conclude quando la ragazza, stanca delle contraddizioni del protagonista, sposa un insegnante di canto presentatole dallo stesso Zeno.

Capitolo VII. Storia di un'associazione commerciale

In questo capitolo il protagonista narra del fallimento dell'impresa gestita insieme a Guido, il marito di Ada. Il fallimento è dovuto a sperperi di denaro da parte di Guido il quale, per cercare di rimediare, inscena un tentato suicidio per riscuotere del denaro dalla moglie in modo da risanare l'azienda. Un secondo tentativo di suicidio, però, si realizza seriamente. Giunto in ritardo al funerale di Guido, Zeno, viene accusato da Ada di aver espresso in tal modo la gelosia verso il defunto marito.

Capitolo VIII. Psicoanalisi

In questo capitolo Zeno riprende a scrivere dopo sei mesi di interruzione ribellandosi al medico e denunciando il suo disprezzo per la psicoanalisi. Proprio in quest' ultima parte, però, Zeno capisce di essere guarito in quanto la malattia interiore di cui soffriva è condivisa da tutta l'umanità ed il fatto di essersene reso conto è segno di salute mentale.

La narrazione si conclude con una sorta di profezia di una catastrofe come metodo di "ritorno alla salute".

L'opera "La coscienza di Zeno" è considerata dalla critica l'ultimo grande capolavoro di I. Svevo. Il testo presenta le tematiche che hanno reso famoso l'Autore triestino, precisamente:

- l'inettitudine: il protagonista, Zeno Cosini, è un inetto per eccellenza; infatti, è privo di determinazione, incapace di prendere delle decisioni.
- la psicoanalisi: tema di fondo dell'opera, riflette il profondo della coscienza del protagonista paziente;
- il "complesso di Edipo": il protagonista ha un rapporto conflittuale con il padre e di rispetto ed affetto con la madre, costituendo il cosiddetto "complesso di Edipo" teorizzato da S. Freud;
- l'ironia: il romanzo contiene molti passaggi ironici inseriti per rendere la lettura di temi "pesanti" più piacevole;
- il pessimismo: la visione dell'uomo che emerge dal testo è senz'altro pessimistica; il culmine si raggiunge nel finale quando emerge il concetto che una catastrofe può risolvere i problemi;
- la morte: è un tema ricorrente nel romanzo e può essere considerato una sfaccettatura del pessimismo;
- la verità: il protagonista si rifugia nella falsità per rendere le situazioni a lui favorevoli; solo alla fine si renderà conto che la verità è la salute, la cura;
- la memoria: l'intero romanzo è basato sul racconto dei ricordi del protagonista.

L'ULTIMA SIGARETTA

Il brano senz'altro più famoso dell'opera "La coscienza di Zeno" (estratto dal terzo capitolo) è quello riportato in tanti testi, anche in "Autori e Opere volume 3A", testo in adozione nella mia Classe, con il titolo "L'ultima sigaretta". Nel brano in questione il protagonista, per adempiere al suggerimento del dottor S. di scrivere un'autobiografia per favorire la cura, descrive in prima persona il suo vizio del fumo

Analisi del testo

Nel testo riportato Zeno Cosini racconta i dettagli del suo vizio per il fumo, sempre combattuto senza successo. Il protagonista ricorda le sue prime esperienze con il fumo, dapprima rese possibili rubando i soldi al padre, poi, una volta scoperto, fumando i resti dei sigari del padre. A vent'anni, il protagonista si accorge di odiare il fumo, causa oltretutto di un suo malessere fisico ma, nonostante la sua condizione, il solo desiderio di fumare un'ultima sigaretta e le proibizioni del padre (*"ma le proibizioni valsero ad eccitarlo"*) eccitavano il suo vizio a tal punto da non essere mai effettivamente l'**ultima** sigaretta. In questo passo si può evincere l'**ossessione** del protagonista per l'ultima sigaretta, la quale sarà dal protagonista stesso considerata superata quando afferma *"da qualche tempo io fumo molte sigarette... che non sono le ultime"*; in questo passo viene infatti affermato che l'ossessione per l'**ultima** sigaretta è superata, ma non il vizio per il fumo. È importante, inoltre, sottolineare la nascita del vizio del fumo, inizialmente come forma di ribellione al padre che fumava senza problemi di fronte a Zeno ma al contempo impediva al figlio di farlo; successivamente, questo vizio si allarga come forma di difesa dalla realtà circostante.

La rivalità nata nel protagonista nei confronti del padre può essere considerata una sfaccettatura del **complesso di Edipo**, riconducibile ad un sentimento di rivalità verso il genitore dello stesso sesso e, non presente in maniera profonda come da definizione ne "La coscienza di Zeno", in un sentimento di amore verso il genitore di sesso opposto.

Questi aspetti sono considerati psicanalitici, seguendo i canoni ideati da S. Freud.

Tra gli altri aspetti rilevanti presenti nel brano ci sono:

- l'inettitudine: il protagonista vorrebbe cambiare il suo destino, ovvero smettere di fumare, ma non riesce a causa della sua mancanza di determinazione, della sua pigrizia, della sua incapacità di prendere delle decisioni.
- l'ironia: nel brano, I. Svevo è ricorso all'utilizzo dell'ironia, ovvero alterando fatti volendo in realtà affermare il contrario in modo da non rendere la lettura monotona; un esempio di ironia può essere individuato nella parte finale, quando Zeno narra di fantasiose date perfette per le sue *ultime* sigarette.
- la memoria: il brano è un racconto di vita del protagonista emerso scavando nella sua memoria al fine di adempiere alla cura psicanalitica prescritta dal dottor S.
- l'autobiografia: il brano è un'autobiografia scritta dal protagonista su consiglio del dottor S. come cura al suo problema psicanalitico.
- l'incertezza dell'uomo: il protagonista è paziente di uno psicanalista perché non riesce ad identificarsi con il mondo circostante.

Le fonti utilizzate nella presente tesi sono state:

- appunti personali
- materiale didattico:
 - libri di testo
 - schede integrative

Sono state inoltre utilizzate fonti informatiche per l'approfondimento disciplinare:

- Il linguaggio SQL: it.wikipedia.org
- L'architettura Client-Server: it.wikipedia.org
- La coscienza di Zeno: oilproject.org, it.wikipedia.org
- Il complesso di Edipo: treccani.it

La versione web della presente tesi è disponibile inoltre all'indirizzo:
nicholasboin.altervista.org (nicholasboin.ml)