Raccolta automatica di dati finanziari dal web
A.A. 2009/2010
Elaborato
ver 1.0

Luca Pellegrini (579306)
Relatore: Federico Filira

28 Settembre 2010

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Stato dell'arte del recupero di informazioni sul web

1.1 Obiettivi

Il Web è una grande fonte di informazione libera e accessibile a tutti. In questa sezione verranno esaminate le problematiche relative all'acquisizione di specifiche informazioni da fonti selezionate per interesse.

Le problematiche che tipicamente si devono affrontare sono:

- valutare e soddisfare il grado di **aggiornamento** delle informazioni: a seconda dell'ambito di applicazione occorre che le informazioni acquisite siano aggiornate con più o meno frequenza;
- occorre sviluppare una **automatizzazione** del processo di acquisizione: ovvero svincolare l'azione di acquisizione delle informazioni dalla presenza dell'operatore umano.

1.2 Information Retrieval

L'information retrieval[1] (IR) è l'insieme delle tecniche utilizzate per il recupero mirato dell'informazione in formato elettronico. Per informazione si intendono tutti i documenti, i metadati, i file presenti all'interno di banche dati o nel world wide web. Il termine è stato coniato da Calvin Mooers alla fine degli anni '40 del Novecento, ma oggi è usato quasi esclusivamente in ambito informatico.
L'IR è un campo interdisciplinare che nasce dall'incrocio di discipline diverse. L'IR coinvolge la psicologia cognitiva, l'architettura informativa, la filosofia, il

[1] Letteralmente: *recupero di informazioni*

design, il comportamento umano sull'informazione, la linguistica, la semiotica, la scienza dell'informazione e l'informatica. Molte università e biblioteche pubbliche utilizzano sistemi di IR per fornire accesso a pubblicazioni, libri ed altri documenti.

Per recuperare l'informazione, i sistemi IR usano i linguaggi di interrogazione basati su comandi testuali. Due concetti sono di fondamentale importanza: query ed oggetto. Le query[2] sono stringhe di parole-chiavi rappresentanti l'informazione richiesta. Vengono digitate dall'utente in un sistema IR (per esempio, un motore di ricerca).
Una tipica ricerca di IR ha come input un comando dell'utente; poi la sua query viene messa in relazione con gli oggetti presenti nella banca dati, e in risposta il sistema fornisce un insieme di record che soddisfano le condizioni richieste.

Spesso i documenti stessi non sono mantenuti o immagazzinati direttamente nel sistema IR, ma vengono rappresentati da loro surrogati. I motori di ricerca del Web come Google e Yahoo sono le applicazioni più note ed ovvie delle teorie di Information Retrieval.

1.3 Modalità di acquisizione

In alcune applicazioni si presenta la necessità di recuperare informazioni aggiornate in modo automatizzato.

Poichè le informazioni devono essere *aggiornate*, occorre ripetere nel tempo l'acquisizione dei dati, con una frequenza che varia in funzione dell'applicazione.
Inoltre per le applicazioni che richiedono un aggiornamento in tempo reale occorre ideare una strategia per non sovraccaricare la fonte con ripetute richieste.

Essendo un'attività ripetitiva, è importante che l'acquisizione venga *automatizzata*. Se le informazioni da acquisire sono disponibili in formato puro, ovvero prive di ridondanza e di formattazione, allora l'acquisizione è immediata. Viceversa se non si dispone di informazioni pure occorre processare il dato in ingresso per interpretarlo ed estrapolare l'informazione di interesse.

1.3.1 Tecnologia push

La tecnologia di tipo *push* descrive una tipologia di comunicazione in internet, dove la richiesta per ciascuna transazione è inizializzata dal publisher (il server centrale). Diversamente, nel modello *pull* la richiesta di trasmissione è inizializzata dal ricevente (il client).

[2] interrogazioni

Uso generale

I servizi push sono spesso basati su preferenze espresse in anticipo. Questo è chiamato modello pubblica/sottoscrivi. Un client può sottoscrivere svariati canali di informazione. Non appena diventa disponibile un nuovo contenuto all'interno di questi canali, il server esegue il push delle informazioni verso gli utenti.
Conferenze sincrone e messaggistica istantanea sono un esempio tipico di servizio push. I messaggi di chat sono inviati all'utente non appena vengono ricevuti dal servizio di messaggistica.
Anche l'email è un sistema di tipo push: il protocollo SMTP sul quale è basata è un protocollo con funzionalità push. L'ultimo passo però (dal mail server al computer desktop) utilizza tipicamente un protocollo di tipo pull, come POP3 o IMAP. I client email moderni fanno sembrare istantanea questa azione grazie a un continuo polling[3] del server mail. Il protocollo IMAP include il comando IDLE, che consente al server di avvisare il client quando arriva un nuovo messaggio.

Implementazioni

HTTP server push È una tecnica per inviare dati da un web server a un web browser. Esistono diversi meccanismi per ottenere il push.
Nella maggior parte delle applicazioni push, il web server evita di terminare la connessione una volta che la risposta è stata recapitata al client. Il web server lascia la connessione aperta in modo che se un evento è ricevuto, esso può inviarlo immediatamente a uno o molteplici client. In caso contrario, una volta chiusa la connessione i dati successivi verrebbero accodati fino alla prossima richiesta del client.
La proposta WHATWG Web Applications 1.0 incluse un meccanismo per fare il push del contenuto verso il client. Ora tale meccanismo è stato standardizzato come parte di HTML5. Un'altra funzionalità relativa ad HTML è l'API WebSockets, che consente la comunicazione tra web server e client tramite una connessione TCP full-duplex.

Java pushlet Una pushlet è una tecnica originariamente sviluppata per le applicazioni web Java, ma può essere utilizzata anche in altri framework web.
In questa tecnica il server trae vantaggio dal connessioni HTTP persistenti e lascia la risposta indefinitivamente aperta (ovvero non la termina mai). Questo comportamento ha l'effetto di ingannare il browser e lo induce a restare continuamente nella modalita di caricamento anche dopo aver terminato la ricezione della pagina effettiva. Il server inoltre invia periodicamente codice javascript per aggiornare il contenuto della pagina, ottenendo così la funzionalità push.

[3] interrogazione

Usando questa tecnica il client non necessita di applet java o plugin per tenere aperta la connessione al server. Un serio problema di questo metodo è la mancanza di qualsiasi forma di controllo del server verso il timing out del client. Si rivela necessario un refresh della pagina ad ogni timeout che occorre dal lato client.

Long polling Questa tecnica è una variazione del tradizionale polling per consentire l'emulazione della tecnologia push.
Con il long polling il client richiede l'informazione al server, similmente a come farebbe con un poll normale. Se il server non dipone di informazioni per il client, invece di inviare una risposta vuota, trattiene la richiesta e aspetta che si rendano disponibili alcune informazioni. Una volta che l'informazione è disponibile (o dopo un certo timeout) viene inviata una risposta al client. Il client solitamente produrrà subito una nuova richiesta per il server, in modo che il server avrà sempre una richiesta in attesa che potrà essere utilizzata per inviare i dati in risposta a un evento.

Limitazioni della tecnologia push

L'utilizzo di una tecnologia di tipo push è possibile solo in seguito a un accordo esplicito tra publisher e client. Per ragioni puramente tecniche, è il publisher stesso che deve essere a conoscenza dei suoi client in modo da poterli avvertire non appena si rende disponibile il contenuto aggiornato.
I servizi di tipo push possono garantire un alto livello di qualità e il perfetto sincronismo tra server e client, pertanto prevedono spesso accordi di natura economica tra colui che produce l'informazione e colui che ne usufruisce.

1.3.2 Tecnologia pull

La tecnologia pull è una tecnica di comunicazione di rete che prevede una richiesta iniziale originata da un client, alla quale risponde un server. Le richieste pull sono il fondamento del network computing, dove molti client richiedono dati da server centralizzati. Il pull è usato intensivamente in internet per le richieste HTTP di pagine web dai siti. Anche moltre altre fonti web, come gli RSS, sono ottenute dal client tramite pull.

1.4 La descrizione dell'informazione: i linguaggi di Markup

Le informazioni nel web sono tipicamente descritte tramite linguaggi di markup.

Un documento si compone di:

- **struttura**: organizzazione logica del documento;
- **contenuto**: parte informativa;
- **presentazione**: aspetto grafico.

Per descrivere e separare le tre componenti di un documento vengono inserite nel documento stesso delle annotazioni (marcatori).
Affinchè sia possibile il trattamento automatico dei documenti, le annotazioni devono seguire regole sintattiche e semantiche ben precise. L'insieme di tali regole definisce il linguaggio di annotazione usato, chiamato linguaggio di markup.

I linguaggi di markup sono linguaggi formali che vengono definiti e realizzati per specificare la struttura e il formato di documenti digitali tramite l'uso di marcatori, chiamati tag.
Un tag è una parola che descrive una porzione del contenuto di un documento. L'insieme dei tag di un linguaggio di markup permette di descrivere la struttura di un documento identificando e separando i componenti logici.
L'utilizzo di linguaggi formali consenste di elaborare in modo automatico il linguaggio scritto.

SGML E' un metalinguaggio per definire linguaggi di markup. Utilizzandone i costrutti è possibile creare un numero infinito di questo tipo di linguaggi.
Un documento SGML[4] può essere facilmente letto sia da un computer che da una persona, a patto che quest'ultima conosca lo standard.
Un esempio di linguaggio derivato da SGML è l'HyperText Markup Language (HTML), ampiamente utilizzato nel web.

HTML, XML, XHTML

- **HTML**: HyperText Markup Language, è un linguaggio di markup che descrive documenti ipertestuali.
- **XML**: eXtensible Markup Language, è un sottoinsieme semplificato di SGML. Permette di definire il proprio formato di markup.
- **XHTML 1.0**: combina la forza di HTML con le potenzialità di XML.

[4] *Standard Generalized Markup Language*

1.5 Tipologie di documenti reperibili nel web

1.5.1 Pagine web

Le pagine web sono tipicamente documenti HTML/XHTML contenenti testo e riferimenti a immagini, suoni e/o contenuti multimediali.

Contenuto testuale

Il testo è presente direttamente all'interno dello stream HTML che compone la pagina. La presenza di diversi tag consente di descrivere la struttura del contenuto informativo, e sono proprio i tag, insieme ai fogli di stile (CSS[5]) a determinare come il browser deve presentare la pagina all'utente.

Contenuto multimediale

All'interno dello stream[6] HTML possono essere presenti riferimenti a file esterni di svariata natura come immagini, suoni e formati multimediali di terze parti. Il browser in questi casi scarica il contenuto aggiuntivo e lo incorpora nella pagina, con le modalità specificate nel sorgente HTML stesso. Tra i formati multimediali più diffusi vi è il formato Flash di Adobe, ormai uno standard de facto. Flash è una tecnologia che permette di creare animazioni vettoriali, anche di notevole complessità. I contenuti Flash vengono interpretati dal browser tramite un apposito plug-in, di cui esiste ovviamente anche un'implementazione prodotta da Adobe stessa.

1.5.2 Web feed

Un web feed è un formato di dati usato per distribuire agli utenti dei contenuti frequentemente aggiornati. I distributori di contenuto forniscono un feed che può essere sottoscritto dagli utenti.
Un tipico scenario di utilizzo di un web feed è il seguente: un content provider[7] pubblica un link al feed sul proprio sito, al quale gli utenti finali possono registrarsi con un programma aggregatore. Per ciascun feed sottoscritto l'aggregatore interroga il server per verificare se sono presenti nuovi contenuti ed eventualmente li scarica.
I web feed sono un esempio di tecnologia pull: gli aggregatori possono essere programmati per controllare periodicamente la presenza di aggiornamenti.

[5] *Cascading Style Sheets*
[6] flusso
[7] fornitore di contenuti

Il tipo di contenuto indicizzato dai web feed è tipicamente HTML (pagine web) o link ad esse.

Un web feed, tecnicamente, è un documento spesso basato su XML. I due formati più diffusi di feed sono RSS e Atom.
I web feed sono concepiti per facilitarne la lettura automatica oltre che quella umana. Ciò significa che i web feed possono essere utilizzati anche per trasferire automaticamente informazioni da un sito all'altro senza l'intervento umano.

1.5.3 RSS

RSS[8] è una famiglia di formati standard di web feed usati per pubblicare contenuti frequentemente aggiornati, compreso testi, audio, video.
Un documento RSS include, oltre al contenuto, dei metadati come le date di pubblicazione e l'autore.
L'utilizzo di un formato XML standard consente di pubblicare l'informazione una sola volta e di visualizzarla su molteplici dispositivi.

1.6 Acquisire i dati contenenti l'informazione: il crawler

Un *web crawler* è un software che naviga il web in modo automatico e metodico. I crawler sono chiamati anche bot, web spider o web robot.
I motori di ricerca, in particolare, usano il crawling come meccanismo per mantenere aggiornati i propri indici. I crawler sono utilizzati per mantenere una copia locale di tutte le pagine visitate, le quali saranno successivamente processate dal motore di ricerca per indicizzarle allo scopo di rendere veloci le ricerche al loro interno.
I crawler possono essere utilizzati anche per acquisire specifiche informazioni da pagine web, come ad esempio raccogliere indirizzi email (solitamente per spam).

1.7 Estrapolare l'informazione: il parsing

Per *parsing* si intende un processo di analisi sintattica di un testo composto da una sequenza di token (ad esempio parole). Il fine di tale processo è la determinazione della struttura grammaticale in riferimento a una grammatica formale.

[8] *Really Simple Syndication*

1.7.1 Parser

In informatica un *parser* è un componente che controlla la sintassi e costruisce una struttura dati di un certo input. Il parser spesso utilizza un analizzatore lessicale separato per creare i token della sequenza di ingresso. I parser possono essere programmati a mano o essere generati semi-automaticamente da qualche tool.

1.7.2 Alcune librerie per il parsing

Vengono qui riportate alcune librerie utilizzate per il parsing in vari linguaggi di programmazione.

- **(PHP) MagpieRSS**: libreria in PHP che fornisce un parser per RSS basato su XML.
- **(Python) Universal Feed Parser**: parsing di feed RSS e Atom in Python.
- **(Ruby) rss.rb**: RSS parser della libreria standard di Ruby.
- **(Python) HTML Parser**: semplice parser per HTML e XHTML, modulo della libreria standard di Python.
- **(Java) HTML Parser**: libreria in java per il parsing di HTML.

Capitolo 2

Stato dell'arte dell'organizzazione delle informazioni storiche

2.1 Obiettivi

Ogni organizzazione è dotata di un sistema informativo, che organizza e gestisce le informazioni necessarie per perseguire gli scopi dell'organizzazione stessa. Un sistema informativo non è necessariamente un sistema automatizzato o digitale, e la sua esistenza prescinde da queste due caratteristiche. In questa sezione si vuole analizzare appunto la parte automatizzata di un sistema informativo e vedere quali sono i principali sistemi per la gestione e la memorizzazione delle informazioni.

2.2 Gestione delle informazioni

Per indicare la parte automatizzata del sistema informativo viene di solito usato il termine sistema informatico.
In quest'ultimo bisogna differenziare il termine dato e informazione. I primi da soli non assumono nessun significato ma, una volta interpretati e correlati opportunamente, forniscono informazioni che consentono di arricchire la nostra conoscenza del mondo:

- **informazione**: notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere;
- **dato**: ciò che è immediatamente presente alla conoscenza, prima di ogni elaborazione; (in informatica) elementi di informazione costituiti da simboli che devono essere elaborati.

Un file consente di memorizzare e ricercare dati, ma fornisce solo semplici meccanismi di accesso e condivisione. In questo modo le procedure scritte in un linguaggio di programmazione sono completamente autonome perché ciascuna di esse utilizza un file "privato". Eventuali dati di interesse per più programmi sono replicati tante volte quanti sono i programmi che li utilizzano. Questo provoca ridondanza[1] e possibilità di inconsistenza[2].
Le base di dati risolvono questi tipi di inconvenienti gestendo in modo integrato e flessibile le informazioni di interesse. Però sono semplicemente una collezione di dati che necessita di essere gestita da un apposito sistema.

2.3 Data Base Management System

Un sistema di gestione di basi di dati è un sistema software in grado di gestire collezioni di dati che siano:

- **grandi**: possono avere dimensioni anche enormi e comunque in generale molto maggiori della memoria centrale disponibile;
- **condivise**: applicazioni e utenti diversi devono poter accedere, secondo opportune modalità, a dati comuni. In questo modo si riduce la ridondanza dei dati, perché si evitano ripetizioni, e conseguentemente si riduce la possibilità di inconsistenza;
- **persistenti**: hanno un tempo di vita illimitato a quello delle singole esecuzioni che le utilizzano.

Chiaramente, in ambito professionale, devono essere garantite anche:

- **affidabilità**: la capacità di conservare intatto il contenuto della base di dati (o di permetterne la ricostruzione) in caso di malfunzionamento hardware/software. I DBMS[3] tramite salvataggio (backup) e ripristino (recovery) gestiscono in modo controllato versioni replicate dei dati;
- **efficienza ed efficacia**: capacità di svolgere le operazioni utilizzando una politica di risparmio di tempo e di occupazione di spazio, garantendo la produttività delle attività degli utenti.

[1] presenza di dati duplicati
[2] presenza di dati che non rispecchiano le informazioni di interesse
[3] *Data Base Managment System*

2.3.1 Modelli Logici

Un modello dei dati è un insieme di concetti utilizzati per organizzare i dati di interesse e descriverne la struttura in modo che essa risulti comprensibile a un elaboratore. Ogni modello dei dati fornisce meccanismi di strutturazione, analoghi ai costruttori di tipo dei linguaggi di programmazione, che permettono di definire nuovi tipi sulla base di tipi (elementari) predefiniti e costruttori di tipo. Le strutture utilizzate da questi modelli, pur essendo astratte, riflettono una particolare organizzazione.

Modello Gerarchico Questo modello nasce alla fine degli anni 60 con l'immissione sul mercato da parte di IBM di IMS (il primo DBMS in assoluto e, appunto, gerarchico). Il nome del modello riflette la struttura sulla quale si appoggia: ogni database è diviso in archivi, a loro volta suddivisi in segmenti (o rami); infine i segmenti sono in relazione tra di loro attraverso legami padre-figlio. Si individua dunque un segmento principale (o radice) dal quale dipendono tutti gli altri segmenti figli. In virtù di questa dipendenza dal padre è possibilie fare dei rifermienti solo passando attraverso la radice, ed inoltre non è possibile, dato un figlio, risalire al padre. Chiaramente questo tipo di architettura, utilizzato per la gestione di grosse moli di dati, non è efficiente in caso di una gestione dinamica dei dati.

Modello Reticolare Il modello reticolare nasce dalla necessità di adattare il modello gerarchico a situazioni più complesse, e per questo c'è chi lo considera come un'estensione di quello gerarchico. La prima differenza consiste nella pluralità di padri che ogni nodo può avere; inoltre per questo modello esistono i normali record, e le correlazioni tra questi vengono espresse attraverso record particolari chiamati record di collegamento (*member*). Oltre ai record (normali e member) c'è una seconda struttura fondamentale chiamata *set* che permette di correlare i record per mezzo di catene di puntatori. Dunque uno schema conterrà dei record collegati da dei set.

Modello Relazionale Il modello relazionale dei dati, sviluppatosi attorno agli anni settanta, è attualmente il più diffuso e permette di definire, per mezzo del costruttore relazione (o stato di relazione o istanza di relazione), l'organizzazione dei dati in insiemi di record (tuple) a struttura fissa. Una relazione viene spesso rappresentata per mezzo di una tabella, in cui le righe rappresentano specifici record (tuple) e le cui colonne corrispondono a campi del record (attributi).

Questo modello si basa su due concetti: relazione e tabella. Infatti, dati due insiemi D1 e D2 si chiama prodotto cartesiano di D1 e D2, l'insieme delle coppie ordinate (v1, v2), tali che v1 è un elemento di D1 e v2 è un elemento di D2. Una *relazione*

matematica sugli insiemi D1 e D2 è un sottoinsieme del prodotto cartesiano di D1 x D2. Queste relazioni possono essere rappresentate graficamente in maniera espressiva sotto forma tabellare. Una tupla a questo punto è definita come:

> *Una tupla su un insieme di attributi X (x rappresenta l'insieme di attributi della relazione) è una funzione t che associa a ciascun attributo A di X un valore del dominio DOM(A)*

La grande potenza di questo modello impone anche un certo grado di rigidà. Infatti per rappresentare in modo semplice la non disponibilità di valori per un dato attributo viene inserito un particolare valore, il *valore nullo*. E' inoltre necessario evitare l'inserimento di dati sbagliati o privi di senso, e per questo esistono dei vincoli ben definiti.
I principali sono:

- Vincolo di dominio
- Vincolo di univocità
- Vincolo di integrità dell'entità
- Vincolo di integrità referenziale

Modello ad Oggetti Lo stile di programmazione moderno tende ad essere sempre più orientato verso la programmazione ad ogetti, e questo rende i DBMS ad Ogetti ideali per programmatori, che possono così sviluppare DBMS come fossero oggetti, e all'occorrenza replicarli o modificarli per crearne di nuovi. L'informazione si è evoluta nel tempo ed oggi, molto più rispetto a qualche anno fa, questa non include solo dei dati ma anche video, grafici, file audio e foto che sono considerati dati complessi. I DBMS relazionali non sono in grado di gestire in maniera efficiente questi dati. Grazie all'integrazione con i linguaggi di programmazione, il programmatote può gestire in un solo ambiente anche il DBMS ad oggetti poiché utilizza lo stesso modello di rappresentazione. Al contrario, utilizzando DBMS relazionali, i programmi che trattano dati complessi dovrebbero essere divisi in due parti: il databse e l'applicativo.

Di tutti i modelli logici elencati quello che più si è diffuso è certamente il modello Relazionale, introdotto da Edgar F. Codd. I DBMS che si appoggiano a questo modello vengono chiamati RDBMS.[4]
Una nota definizione di ciò che costituisce un RDBMS è data dalle 12 regole di

[4] *Relational Database Management System*

Codd. Tuttavia molte delle prime implementazioni del modello relazionale non erano conformi a tali regole, per cui il termine venne gradualmente cambiato fino a descrivere una più ampia classe di sistemi di basi di dati.

I requisiti minimi perchè un sistema venisse riconosciuto come RDBMS erano:

- deve presentare i dati all'utente sotto forma di relazioni (una rappresentazione a tabelle può soddisfare questa proprietà)
- deve fornire operatori relazionali per manipolare i dati in forma tabellare

2.3.2 Livelli di astrazione nei DBMS

Esiste un'architettura standardizzata articolata su tre livelli e per ognuno di questi esiste uno schema:

- **schema logico**: descrizione dell'intera base di dati per mezzo del modello logico adottato dal DBMS (modello relazionale, modello gerarchico, modello reticolare e modello a oggetti);
- **schema fisico o interno**: rappresentazione dello *schema logico* utilizzato per mezzo di strutture fisiche di memorizzazione;
- **schema esterno**: descrizione di una porzione della base di dati di interesse per mezzo del modello logico. Uno schema esterno può prevedere organizzazioni dei dati diverse rispetto a quelle utilizzare nello *schema logico*. Quindi è possibile associare a uno *schema* logico vari schemi esterni.

2.3.3 Indipendenza dei dati

L'architettura a tre livelli garantisce un'essenziale qualità ai DBMS: l'indipendenza dei dati. Questa può essere suddivisa in:

- **indipendenza fisica**: consente di interagire con il DBMS in modo indipendente dalla struttura fisica dei dati. Si possono così modificare le strutture fisiche (per esempio la modalità di gestione dei file) senza influire sulle descrizioni ad alto livello e quindi sui programmi che utilizzano i dati stessi;
- **indipendenza logica**: consente di interagire con il livello esterno della base di dati in modo indipendente dal livello logico. Per esempio è possibile aggiungere uno schema esterno senza dover modificare lo schema logico e la sottostante organizzazione fisica dei dati.

Gli accessi alla base di dati avvengono solo attraverso il livello esterno (che può coincidere con quello logico). È il DBMS a tradurre queste operazioni per i livelli sottostanti.

2.3.4 Linguaggi

Esistono diversi tipi di linguaggi che consentono di interagire con il DBMS e si distinguono in base alle loro funzioni; le due grandi categorie sono:

- **Data Definition Language** (*DDL*): linguaggi di definizione dei dati, utilizzati per definire gli schemi logici, esterni e fisici e e le autorizzazioni per l'accesso.
- **Data Manipulation Language** (*DML*): linguaggi di manipolazione dei dati, utilizzati per l'interrogazione e l'aggiornamento delle istanze di basi di dati.

Structured Query Language È un linguaggio strutturato di interrogazione *completo* perchè permette sia la definizione dei dati (DDL) che la manipolazione (DML) attraverso aggiornamenti ed interrogazioni. Questo linguaggio è stato progettato per gestire dati in un sistema di tipo relazionale. Consente di creare e modificare schemi di database, oltre a permettere l'utilizzo dei dati e la gestione degli strumenti di controllo e di accesso.
SQL utilizza dei costrutti di programmazione denominati *query* per le interrogazioni.
Creato da IBM negli anni settanta per gestire il database relazionale da loro brevettato, inzialmente prese il nome di Sequel, e nel 1986 l'ANSI[5] lo standardizzò con la sigla SQL-86.
La maggior parte delle implementazioni ha come interfaccia la classica linea di comando per l'esecuzione dei comandi, in alternativa all'interfaccia grafica.

2.3.5 Implementazioni attuali

Esistono diversi DBMS attualmente utilizzati dalle aziende di tutto il mondo. Alcuni di questi sono proprietari, altri sono di tipo Open Source. Di seguito l'elenco dei più diffusi:

DBMS proprietari

- **IBM DB2**: DB2 è un RDBMS della IBM. La sua prima versione risale al 1983 e secondo molti è stato il primo prodotto a utilizzare il linguaggio SQL.

[5] *American National Standard Institute*

- **Microsoft SQL Server**: Microsoft SQL Server è un RDBMS prodotto da Microsoft. Nelle prime versioni era utilizzato in prevalenza per basi dati medio-piccole, ma a partire dalla versione 2000 ha preso piede anche per la gestione di basi dati di grandi dimensioni.
- **Microsoft Access**: Microsoft Access è un RDBMS realizzato da Microsoft, incluso nel pacchetto Microsoft Office Professional ed unisce il motore relazionale Microsoft Jet Database Engine ad una interfaccia grafica.
- **Oracle**: Oracle è uno tra i più famosi RDBMS. La prima versione di Oracle risale al 1977, da allora sono state introdotte numerose modifiche e miglioramenti per seguire gli sviluppi tecnologici.

Attualmente DB2 e Oracle si contendono il primo posto nel mercato dei DBMS.

DBMS open source o free software

- **MySQL**: MySQL è un RDBMS composto da un client con interfaccia a caratteri e un server, entrambi disponibili sia per sistemi Unix come GNU/-Linux che per Windows, anche se prevale un suo utilizzo in ambito Unix.
- **PostgreSQL**: PostgreSQL è un completo database relazionale ad oggetti rilasciato con licenza libera (stile Licenza BSD[6]). PostgreSQL è una reale alternativa sia rispetto ad altri prodotti liberi come MySQL e Firebird SQL che a quelli a codice chiuso come Oracle o DB2. Offre caratteristiche uniche nel suo genere che lo pongono per alcuni aspetti all'avanguardia nel settore dei database.
- **Firebird SQL**: Firebird SQL è un RDBMS opensource distribuito sotto licenza IPL[7]. Supporta numerosi sistemi operativi e le principali caratteristiche di questo prodotto sono l'alto livello di conformità con gli standard SQL, la completa integrazione con molti linguaggi di programmazione e la facile installazione e manutenzione del software.
- **SQLite**: SQLite è una libreria software scritta in linguaggio C che implementa un DBMS SQL di tipo ACID incorporabile all'interno di applicazioni. È stato rilasciato nel pubblico dominio dal suo creatore, D. Richard Hipp. SQLite permette di creare una base di dati (comprese tabelle, query, form, report) incorporata in un unico file, come nel caso dei moduli Access di Microsoft Office e Base di OpenOffice.org.

[6] *Berkeley Software Distribution*
[7] *Interbase Public License*

2.4 Criteri di scelta e DBMS a confronto

Per decidere quale sia il DBMS più adatto ad ogni esigenza è necessario analizzare diversi aspetti, e non solo quelli di natura meramente economica.
Una volta definite le caratteristiche di interesse ogni DBMS preso in considerazione verrà valutato secondo una scala predeterminata.
La valutazione potrebbe quindi basarsi sulle funzionalità integrate, il tipo di licenza, la compatibilità con il sistema operativo, i tipi di dato supportati, la capacità di supportare oggetti esterni, la sicurezza, le caratteristiche e funzionalità dei tool di supporto.
Chiaramente la scelta finale ricadrà sul pacchetto che ha ottenuto il punteggio maggiore.

Per comporre la valutazione appena descritta si utilizzano delle tabelle, come le seguenti, che mettono in risalto le qualità dei diversi DBMS.

DBMS	**Costruttore**	**Ultima versione stabile**	**Licenza**
DB2	IBM	9.7 (22/04/2009)	Proprietaria
Microsoft Access	Microsoft	14 (2010)	Proprietaria
MySQL	Sun Microsystems	5.1.46 (06/04/2010)	GPL
Oracle	Oracle Corp.	11g Rel.2 (10/2009)	Proprietaria
PostgreSQL	PostgreSQL GDG[8]	8.4.4 (17/05/2010)	Free and Open Source
SQLite	D. Richard Hipp	3.6.22 (06/01/2010)	Dominio Pubblico

Tabella 2.1: Informazioni generali

DBMS	**Windows**	**Mac OS X**	**Linux**	**Unix**
DB2	Sì	Sì	Sì	Sì
Microsoft Access	Sì	No	No	No
MySQL	Sì	Sì	Sì	Sì
Oracle	Sì	Sì	Sì	Sì
PostgreSQL	Sì	Sì	Sì	Sì
SQLite	Sì	Sì	Sì	Sì

Tabella 2.2: Compatibilità Sistemi operativi

DBMS	**Dimensione Max DB**	**Dimensione Max tabella**	**Dimensione Max riga**	**Colonne per riga (max)**
DB2	512 TB	512 TB	32,677 B	1012
Microsoft Access	2 GB	2 GB	16 MB	256
MySQL	Illimitata	256 TB	64 KB	4096
Oracle	Illimitata	4 GB * dimensione blocco	8 KB	1000
PostgreSQL	Illimitata	32 TB	1.6 TB	250-1600
SQLite	32 TB	N.D.	N.D.	32767

Tabella 2.3: Limiti delle dimensioni dei dati

DBMS	**Network Encryption**	**Regole complessità password**	**Separazione ruoli utente**	**Certificato di sicurezza**
DB2	Sì	Sì	Sì	Sì (EAL4+)
Microsoft Access				
MySQL	Sì (SSL 4.0)	No	No	Sì
Oracle	Sì	Sì	Sì	Sì (EAL4+)
PostgreSQL	Sì	No	No	Sì (EAL1)
SQLite	No (Solo permessi file)	No	No	No

Tabella 2.4: Sicurezza e controllo degli accessi

Capitolo 3

Scenario di sviluppo con specifiche funzionali

3.1 Obiettivi

Questo elaborato consiste nello sviluppo di un modulo applicativo in grado di cercare autonomamente informazioni e dati da fonti web, memorizzarle su un database, e renderle disponibili all'utente attraverso un'interfaccia grafica. I dati in questione sono di tipo finanziario e permettono di seguire e l'andamento dei titoli di borsa di interesse per l'utente. Oltre alla memorizzazione giornaliera dei dati, vengono creati dei grafici che permettono una visualizzazione ed una comprensione istantanea dei dati raccolti. La ricerca e la memorizzazione dei dati avviene ogni quindici minuti e la storicizzazione interesserà solamente gli ultimi dati raccolti che verranno memorizzati nel database e permetteranno di fare delle analisi a lungo termine.

3.2 Tipologie di dati

Per ogni titolo inserito dall'utente il sistema memorizzerà:

- **Rating**
- **Rendimento** netto o lordo in base al dettaglio dei dati
- **Variazione** (in percentuale), minimi e massimi
- **Valore Monetario**
- **Scadenza investimento**

3.2.1 Rating

Il rating è un metodo utilizzato per classificare sia i titoli obbligazionari che le imprese in base alla loro rischiosità. In questo caso, essi si definiscono rating di merito creditizio da non confondersi ai rating etici. Viene espresso attraverso un voto in lettere (vedi tabelle sotto), che stabilisce la capacità di una data azienda di ripagare il debito contratto. In base al rating il mercato stabilisce un premio per il rischio di cui un'azienda si fa carico accettando quel determinato investimento. Scendendo nel rating aumenta il premio per il rischio e quindi l'emittente deve pagare uno spread maggiore rispetto al tasso risk-free. I rating sono periodicamente pubblicati da agenzie specializzate, principalmente Standard & Poor's, Moody's e Fitch Ratings. Per avere un rating, una società, una banca o uno Stato devono rivolgere una richiesta esplicita a una delle agenzie di rating. Questo servizio è a pagamento, e per questo è chiaro come ci sia un conflitto di interessi visto che non raramente, la maggior fonte di finanziamento dei costosi studi che valutano il rating, non sono le agenzie di stampa e la comunità finanziaria, ma le stesse società emittenti oggetto dell'indagine e singoli investitori con molta liquidità. Terminato il lavoro dell'analista, entra in azione un comitato. Sarà, infatti, un organo collegiale - e non un singolo analista - a valutare tutto il materiale raccolto e ad esprimere un giudizio sotto forma di rating. In seguito, il rating viene votato a maggioranza dal comitato, formato da esperti del settore in cui opera la società che si sta valutando.

AAA	Elevata capacità di ripagare il debito
AA	Alta capacità di pagare il debito
A	Solida capacità di ripagare il debito, che potrebbe essere influenzata da circostanze avverse
BBB	Adeguata capacità di rimborso, che però potrebbe peggiorare
BB, B	Debito prevalentemente speculativo
CCC, CC	Debito altamente speculativo
D	Società insolvente

Tabella 3.1: Classi di rating Standards & Poor's

Aaa	Livello minimo di rischio
Aa	Debito di alta qualità
A	Debito di buona qualità ma soggetto a rischio futuro
Baa	Grado di protezione medio
Ba	Debito con un certo rischio speculativo
B	Debito con bassa probabilità di ripagamento
Caa, Ca, C	Società insolvente

Tabella 3.2: Classi di rating Moody's

3.2.2 Rendimento, variazione e valore monetario

Rendimento Il rendimento finanziario (in inglese yield), è l'utile di un investimento espresso in percentuale. Il rendimento **alla scadenza** di un investimento rappresenta il rendimento che l'investitore otterrebbe mantenendo il titolo acquistato in portafoglio fino alla sua naturale scadenza. il rendimento di una azione dipende dall'incremento (o decremento) del valore dell'azione in un dato periodo a cui si aggiungono eventuali dividendi pagati nello stesso periodo. Se $\mathcal{P}_1$ è il valore di vendita di un'azione, $\mathcal{P}_0$ il prezzo di acquisto ($\mathcal{P}_1$ significa prezzo al momento 1 e $\mathcal{P}_0$ prezzo al momento 0) e $\mathcal{D}$ il dividendo complessivamente pagato dall'azione tra il momento 0 e il momento 1, il rendimento percentuale può essere calcolato come:

$$\frac{\mathcal{P}_1 + \mathcal{D}}{\mathcal{P}_0}$$

Tra gli azionisti, possiamo individuare due grandi categorie: i cassettisti e gli speculatori.
I cassettisti tendono a tenere le azioni in portafoglio per lunghi periodi, generalmente poiché sono interessati a diritti di natura amministrativa (come il diritto di voto); a questa categoria di azionisti preme quindi soprattutto prevedere l'entità dei dividendi futuri.
Gli speculatori, al contrario, non sono interessati ai diritti amministrativi e mantengono in portafoglio le azioni per un breve arco di tempo, aspettando che il loro prezzo salga abbastanza per permettere loro di realizzare una plusvalenza; se consideriamo poi che il breve tempo di detenzione delle azioni spesso non permette loro nemmeno di percepire i dividendi, il loro interesse si concentrerà sul prezzo dell'azione.
Quello descritto fino adesso è però il cosiddetto rendimento *lordo*. Per calcolare quello netto, è necessario detrarre tutte le spese accessorie e, chiaramente, le imposte.

Variazioni Con questo termine si indicano i cambiamenti del prezzo (*valore monetario*) di un titolo che si registrano ogni giorno in borsa. Questi cambiamenti sono dovuti a diversi fattori, quali lo stato di salute patrimoniale dell'azienda o più semplicemente la prevalenza all'aquisto o alla vendita delle azioni di un dato titolo. Questo indice può essere espresso in punti (euro) oppure in punti percentuali. Esiste inoltre un'indice più generale che tiene conto di tutte le transazioni di una borsa, e che riporta l'andamento generale delle transazioni.

Valore Monetario E' il valore dell'azione espresso nella valuta locale. Questo è un parametro molto importante per gli investitori speculatori, che fondano il loro guadagno sulla compravendita di azioni, e quindi sulle plusvalenze, e per fare questo ne tengono sempre sotto osservazione il prezzo.

3.2.3 Pacchetti completi e Asset Allocation

Le banche offrono dei pacchetti completi per gli investitori che non intendono seguire gli andamenti dei singoli titoli. Questi pacchetti sono spesso formati sia da azioni, che da obbligazioni, e la banca fornisce all'investitore l'**Asset Allocation** che non è altro che il riepilogo della struttura del pacchetto. La banca informa il cliente dell'andamento del pacchetto, e per questo spesso risulta difficile reperire in rete questa informazione. Il modulo permetterà l'inserimento manuale di tutti i dati riguardanti i pacchetti ed automaticamente fornirà i dati riguardanti gli indici generali di borsa (azionari e obbligazionari).

Asset Allocation

	% Lunga	% Corta	% Netta
Azioni	20,54	4,81	15,73
Obbligazioni	14,59	9,35	5,24
Liquidità	73,19	0,80	72,38
Altro	6,66	0,00	6,66

Figura 3.1: Descrizione del pacchetto con Asset Allocation

3.3 Frequenza dei rilevamenti

I dati giornalieri verranno rilevati automaticamente dalle ore 9.30 alle 17.30 ogni 15 minuti. Questo solo dal Lunedì al Venerdì. Inoltre, l'ultima rilevazione di

ogni giorno è quella che verrà definitivamente memorizzata nello storico. I dati giornalieri invece vengono continuamente sostituiti da dati più recenti e hanno il solo scopo di definire nel dettaglio l'andamento a breve termine del titolo.

3.4 Dimensione dell'archivio di storicizzazione

Il database cresce linearmente dal momento dell'accensione del sistema. Per evitare problemi di spazio, è possibile azzerare il database e copiare tutto lo storico attuale in un'altro database, in modo tale da peremettere all'utente il ripristino dello spazio necessario al funzionamento del modulo.

3.5 Funzionalità del sistema

Oltre alla ricerca e alla memorizzazione dei dati, il modulo permette l'analisi dei dati attraverso un'interfaccia grafica. Grazie a questa l'utente potrà visualizzare i dati giornalieri di ogni titolo e, mediante dei grafici, anche l'andamento sul lungo termine con diverse scale (settimanale, mensile, annuale).

Capitolo 4

Analisi e ranking delle fonti esistenti

In relazione a quanto esposto nel Capitolo 1, è necessario fare un'analisi delle fonti web esistenti secondo diversi criteri quali la completezza e la qualità dei dati, la facilità di estrazione dei dati e la frequenza di aggiornamento. Non è da escludere l'ipotesi di utilizzare delle fonti a pagamento, nel caso in cui quelle gratutite si verificassero inadegute. E' necessario che la fonte web scelta sia *completa*, cioè che metta a disposizione tutti i dati descritti nel Capitolo 3 (eccezzion fatta per il rating e per la data di fine investimento)

4.1 Fonti principali

I principali siti italiani che trattano i mercati finanziari sono:

- Milano Finanza
- Il sole 24 Ore
- Borsa Italiana
- Soldionline
- Finanza online

Seguono delle brevi descrizioni dei tre siti più frequentati.

4.1.1 Milano Finanza

Figura 4.1: Sito Milano Finanza

Sicuramente Milano Finanza (di Class editori) è una valida alternativa all'ormai diffusissimo Sole 24 Ore. Milano Finanza offre un sito aggiornatissimo e ricco di contenuti che si riesce ad integrare molto bene con l'edizione cartacea. Molto interessante l'inserto Web & Week-end di cui è possibile trovare sia l'edizione on-line che quella cartacea. In sostanza Milano Finanza si propone di rompere il monopolio dell'informazione economica e finanziaria creata dal colosso della Confindustria (Il Sole 24 Ore). Inoltre è disponibile un'applicazione per iPad che permette di visualizzare i contenuti del giornale e di accedere a tutte le altre notizie, commenti ed analisi dei mercati.

4.1.2 Il sole 24 ore

Figura 4.2: Sito Il Sole 24 Ore

Il sito di Confindustria, è ormai senza dubbio il più conosciuto e consultato in italia. Sono presenti diverse sezioni che spaziano dall'ambito economico/finanziario a quello tributario. Per quanto riguarda la sezione finanziaria è possibile seguire l'andamento di azioni, obbligazioni, tassi e materie prime. Nella sezione dedicata alle Azioni è visualizzato, attraverso dei grafici, l'andamento dei titoli generali delle diverse borse mondiali (FTSE MIB, DAX 30, DJIA, NASDAQ). Sia per la parte azionaria, che per quella obbligazionaria è possibile seguire l'andamento dei titoli attraverso grafici e tabelle. Inoltre, previa registrazione, si può usufruire di alcuni strumenti (Portafoglio e Listino personale) per la gestione di alcuni titoli scelti dall'utente.

4.1.3 Borsa Italiana

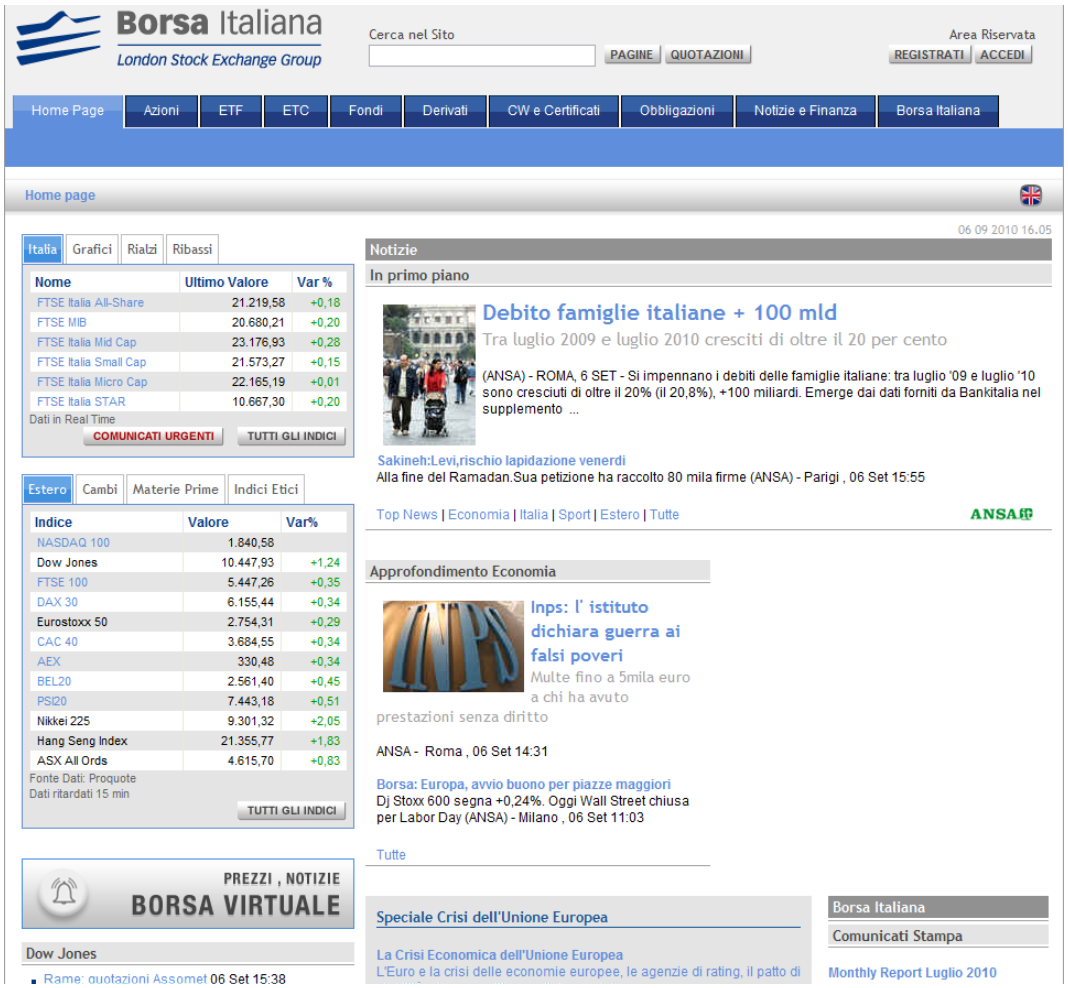


Figura 4.3: Sito Borsa Italiana

E' il sito ufficiale della Borsa Italiana, integrata dal 2007 con il Gruppo London Stock Exchange. Si tratta di un sito bello sia da un punto di vista grafico ed estetico, che da un punto di vista pratico: la home page contiene un gran numero di risorse, decisamente interessanti, se non indispensabili per il risparmiatore e l'investitore. Vasto il panorama informativo, che comprende le news, il mercato giorno per giorno, i mercati internazionali e le società quotate, i dati e le statistiche, i servizi on line. Come per il Sole 24 Ore, anche qui troviamo diverse sezioni che permettono di seguire nel dettaglio sia azioni che obbligaizioni. La ricerca delle azioni può essere fatta attraverso il nome, mentre per le obbligazioni è necessario inserire il codice ISIN.

4.2 Ranking e scelta delle fonti

La scelta delle fonti si basa su alcuni aspetti che riguardano i dati e la loro qualità:

- **Completezza** Per completezza si intende la quantità di informazioni presenti, ma relativamente alle sole informazioni di interesse. Una fonte può essere considerata completa se rende disponibili tutti i dati che si è scelto di rilevare nel terzo capitolo.
- **Attendibilità** Per attendibilità si intende la qualità dell'informazione presente, intesa come il grado di corrispondenza con il dato reale. Una fonte può essere considerata tanto più attendibile quanto più i suoi dati sono vicini al dato reale.
- **Formato dei dati** Come formato dei dati si intende valutare la facilità con cui le informazioni pubblicate sono acquisibili dal sistema.

Sito Web	Completezza	Attendibilità	Formato dati
Milano Finanza	+	+	+
Il sole 24 Ore	++	++	-
Borsa Italiana	++	++	++
Soldionline	+	-	++
Finanza online	-	+	+

Tabella 4.1: Ranking delle fonti

Le valutazioni date si basano su quanto riportato nei capitoli 1 e 3, e permettono di scegliere coerentemente la fonte da utilizzare. I risultati della tabella definiscono:

1. Borsa Italiana
2. Milano Finanza
3. Il sole 24 ore

La fonte scelta è il sito **Borsa Italiana**.

Poiché il rating è un servizio a pagamento erogato solo da centri specializzati (Moody's, Standards & Poor's, ...), non tutte le aziende lo richedono e, conseguentemente, non tutti i titoli sono valutati. Vista la bassa frequenza di aggiornamento e la difficoltà nel reperirle automaticamente dal web, le informazioni riguardanti il rating di ogni titolo, se esistenti, saranno aggiornate manualmente dall'utente.

Capitolo 5

Progettazione di un crawler per l'acquisizione automatica dei dati

5.1 Obiettivi

Come già illustrato in precedenza, il modulo dovrà acquisire automaticamente i dati da fonti web prestabilite. Il compito della raccolta viene svolto da un cosiddetto **crawler** che, una volta estrappolati i dati dalle pagine web, dovrà inserirli nel database per permetterne una successiva consultazione.

5.2 Analisi della struttura della fonte

Per le ragioni esposte nel capitolo 4, il sito scelto è www.borsaitaliana.it. La pagina è composta da diverse parti; per la raccolta dei dati è necessario scansionare le due tabelle (quella centrale e quella di destra).

Fiat

Ultimo Prezzo	Var %	Data - Ora Ultimo Contratto	Fase di Mercato
10,29	-1,44	16/09/10 - 15.53.47	Negoziazione Continua

Dati ritardati di 15 minuti

Scheda | Grafico | Contratti | Dati Completi | Analisi Tecnica | Notizie

Ultimi Prezzi	No	Numero Proposte	Quantità Acquisto	Prezzo Acquisto	Prezzo Vendita	Quantità Vendita	Numero Proposte
10,29	1	4	15.604	10,28	10,29	39.318	22
10,29	2	20	61.976	10,27	10,30	83.135	34
10,28	3	22	56.591	10,26	10,31	40.195	18
10,28	4	22	56.083	10,25	10,32	47.914	20
10,29	5	17	40.069	10,24	10,33	27.809	7

Codice Isin:	IT0001976403
Codice Di Negoziazione:	F
Settore:	Automobili E Componentistica
Mercato/Segmento:	MTA
Cap Sociale:	5.461.237.425
Capitalizzazione:	11.372.590.039
Lotto Minimo:	1,00
Fase di Mercato:	Negoziazione Continua
Prezzo Ultimo Contratto:	10,29
Var %:	-1,44
Var Assoluta:	-0,15
Pr Medio Progr:	10,298
Data - Ora Ultimo Contratto:	16/09/10 - 15.53.47
Quantità Ultimo:	260
Quantità Acquisto:	9.614
Prezzo Acquisto:	10,28
Prezzo Vendita:	10,29
Quantità Vendita:	840
Quantità Totale:	21.140.426
Numero Contratti:	11.843

Controvalore:	217.711.418,86
Max Oggi:	10,47
Max Anno:	11,04 - 12/01/10
Min Oggi:	10,15
Min Anno:	7,525 - 16/02/10
Chiusura Precedente:	10,44
Prezzo di riferimento:	10,44 - 15/09/10 17.40.00
Prezzo ufficiale:	10,41213 - 15/09/10 17.40.00
Apertura Odierna:	10,37
Performance 1 Mese %:	+8,32
Performance 6 Mesi %:	+10,23
Performance 1 Anno %:	+16,73

Figura 5.1: Pagina Dati Completi del sito Borsa Italiana

5.2.1 Borsa Italiana

Questo sito presenta in due sezioni separate le azioni e le obbligazioni. Tutti i dati sono riportati in forma testuale, e sono formattati in modo da presentarsi sotto forma tabellare.

Codice Isin:	IT0001976403
Codice Di Negoziazione:	F
Settore:	Automobili E Componentistica
Mercato/Segmento:	MTA
Cap Sociale:	5.461.237.425
Capitalizzazione:	10.648.976.080
Lotto Minimo:	1,00
Fase di Mercato:	Chiusura
Prezzo Ultimo Contratto:	9,38
Var %:	-2,24
Var Assoluta:	-0,215
Pr Medio Progr:	9,445
Data - Ora Ultimo Contratto:	20/08/10 - 17.30.18
Quantità Ultimo:	790.781
Quantità Acquisto:	
Prezzo Acquisto:	
Prezzo Vendita:	
Quantità Vendita:	
Quantità Totale:	13.603.736
Numero Contratti:	8.432

(a) Tabella principale

Controvalore:	220.194.266,34
Max Oggi:	10,40
Max Anno:	11,04 - 12/01/10
Min Oggi:	10,24
Min Anno:	7,525 - 16/02/10
Chiusura:	10,38
Prezzo di riferimento:	10,38 - 14/09/10 17.40.00
Prezzo ufficiale:	10,33342 - 14/09/10 17.40.00
Apertura Odierna:	10,28
Performance 1 Mese %:	+9,26
Performance 6 Mesi %:	+10,90
Performance 1 Anno %:	+18,02

(b) Tabella secondaria

Figura 5.2: Tabelle per la raccolta dei dati

Il link utilizzato per la raccolta dei dati è quello della sezione *Dati Completi*, sia per i titoli azionari che per quelli obbligazionari.

`http://www.borsaitaliana.it/borsa/azioni/dati-completi.html?isin=IT0001976403&lang=it`

`http://www.borsaitaliana.it/borsa/obbligazioni/mot/obbligazioni-in-euro/dati-completi.html?isin=IT0003022701&lang=it`

Come è facilmente intuibile dai link, l'unico parametro necessaria e che stabilisce la pagina è il codice *ISIN*, che identifica univocamente un titolo, indipendentemente dal tipo (azione, obbligazione).

5.2.2 Codice HTML

Il codice che segue descrive le prime otto righe della tabella principale (a) riportata nella figura precedente. Una volta analizzato il codice sorgente delle pagine dei titoli di interesse, è necessario fare il parsing del codice HTML della pagina per estrappolare le informazioni necessarie.

```
<tr class="odd"> <td class="name" scope="row">Codice Isin:</td>
   <td class="name">IT0001356887</td> </tr>

<tr class="even"> <td class="name" scope="row">Codice Di
   Negoziazione:</td> <td class="name">KRE</td> </tr>

<tr class="odd"> <td class="name">Settore:</td> <td class="name"
   >Tecnologia</td> </tr>

<tr class="even"> <td class="name">Mercato/Segmento:</td> <td
   class="name">MTA</td> </tr>

<tr class=odd> <td class="name">Cap Sociale:</td> <td class="
   name">44.411.705</td> </tr>

<tr class=even> <td class="name">Capitalizzazione:</td> <td
   class="name">70.826.511</td> </tr>

<tr class=odd> <td class="name" scope="row">Lotto Minimo:</td> <
   td class="name">1,00</td> </tr>

<tr class=even> <td class="name" scope="row">Fase di Mercato:</
   td> <td class="name">Negoziazione Continua</td> </tr>
```

5.3 Progettazione crawler

Per la progettazione del crawler è stato scelto di utilizzare, come linguaggio di programmazione, python. Per questo linguaggio sono presenti in rete alcune librerie per facilitare le operazioni di parsing. Per questo modulo sono state utilizzate quelle denominate Beautiful-soup.

5.3.1 Linguaggio di programmazione: Python

Figura 5.3: Logo di Python

Python è un linguaggio di programmazione ad alto livello, rilasciato pubblicamente per la prima volta nel 1991 dal suo creatore Guido van Rossum, programmatore olandese attualmente operativo in Google. Attualmente, lo sviluppo di Python viene gestito dall'organizzazione no-profit Python Software Foundation.

Supporta diversi paradigmi di programmazione, come quello object-oriented (con supporto all'ereditarietà multipla), quello imperativo e quello funzionale, ed offre una tipizzazione dinamica forte. È fornito di una libreria built-in estremamente ricca, che unitamente alla gestione automatica della memoria e a robusti costrutti per la gestione delle eccezioni fa di Python uno dei linguaggi più ricchi e comodi da usare.

Python è un linguaggio pseudocompilato: un interprete si occupa di analizzare il codice sorgente (semplici file testuali con estensione .py) e, se sintatticamente corretto, di eseguirlo. In Python, non esiste una fase di compilazione separata che genera un file eseguibile partendo dal sorgente. L'esser pseudointerpretato rende Python un linguaggio portabile. Una volta scritto un sorgente, esso può essere interpretato ed eseguito sulla gran parte delle piattaforme attualmente utilizzate, con l'univo vincolo della presenza della versione corretta dell'interprete.

La sua sintassi è pulita e snella così come i suoi costrutti, decisamente chiari e non ambigui. I bocchi logici vengono costruiti semplicemente allineando le righe allo stesso modo, incrementando la leggibilità e l'uniformità del codice anche se vi lavorano diversi autori.

Infine, Python è free software: non solo il download dell'interprete per la propria piattaforma, così come l'uso di Python nelle proprie applicazioni, è completamente gratuito; ma oltre a questo Python può essere liberamente modificato e così ridistribuito, secondo le regole di una licenza pienamente open-source.

5.3.2 Librerie per il parsing HTML/XML

Come già accennato, per facilitare le operazioni di parsing, si utilizzano delle librerie che permettono di utilizzare delle funzioni che consentono di navigare nel codice normalizzato del sito di interesse.

Software	Licenza	Piattaforma	Versione Python
Beautiful Soup	Python License	qualsiasi (Python puro)	2.3-2.6/3
Mechanize	BSD	qualsiasi (Python puro)	2.4-2.7

Tabella 5.1: Librerie per il parsing di HTML/XML

In questo particolare caso si è scelto di utilizzare **BeautifulSoup**.

5.3.3 Beautiful Soup

Beautiful Soup è una libreria per il parsing HTML/XML per Python. Tre sono le principali caratteristiche che lo rendono molto efficiente:

1. BS[1] riescie a gestire eventuali cattivi markup. Infatti restituisce un "parse-tree" che rappresenta il documento originale e che viene analizzato per raccogliere i dati di interesse.

2. BS fornisce pochi e semplici metodi ed idiomi propri della programmazione in Python per la navigazione, ricerca e modifica del "parse-tree": sostanzialmente è un toolkit per il sezionamento del documento e per l'estrazione dei dati di interesse. Non è dunque necessario creare un parser personalizzato per ogni applicazione.

3. BS converte automaticamente i documenti che riceve in Unicode, e quelli che restituisce in UTF-8. Non è necessario predefinire la codifica, salvo per i documenti nei quali non sia specificata la codifica e BS non sia in grado di riconoscerla automaticamente. In tal caso basta semplicemente specificare la codifica originale.

Beautiful Soup è praticamente in grado di fare il parsing di qualunque cosa. É possibile, ad esempio, trovare tutti i link (Trova tutti i link), oppure di fare delle ricerche più accurate (Trova tutti i link che sono uguali a foo.com). Operazioni che normalmente richiederebbero ore di lavoro, con Beautiful Soup richiedono solo pochi minuti.

[1] *Beautiful Soup*

5.3.4 Panoramica del codice

Il seguente codice è la parte del crawler che esegue la ricerca e il parsing delle informazioni dalla pagina del sito Borsa Italiana. In seguito si vedrà anche la parte che, invece, provvede ad inserire le informazioni raccolte nel database.

```
conn = psycopg2.connect("dbname=Finanza user=crawler")
cur_search = conn.cursor()
cur_modify = conn.cursor()
cur_search.execute("SELECT codice, tipo FROM investimento WHERE
    tipo='a' OR tipo='o';")
for row in cur_search:
    isin = row[0]
    tipo = row[1]
    if tipo=="a":
        source = "http://www.borsaitaliana.it/borsa/azioni/dati-
            completi.html?isin="+isin+"&lang=it"
        title = "Dati Finanziari Completi del Titolo "
    else:
        source = "http://www.borsaitaliana.it/borsa/obbligazioni
            /mot/obbligazioni-in-euro/dati-completi.html?isin="+
            isin+"&lang=it"
        title = "Dati Finanziari Completi del Obbligazione in
            Euro "
    file = urllib.urlopen(source)
    soup = BeautifulSoup(file)
    titolo = soup.contents[3].contents[1].contents[1].contents
        [0].replace('\t','').replace(title,'').replace(' - Borsa
        Italiana','').replace('\n','').replace('\r','')

    table1 = soup.findAll('tbody')[2]
    table2 = soup.findAll('tbody')[3]

        if tipo=="a":
        prezzo = table1.contents[17].contents[3].contents[0].
            replace(',','.')
        variazione = table1.contents[19].contents[3].contents
            [0].replace(',','.')
        assoluta = table1.contents[21].contents[3].contents[0].
            replace(',','.')
        massimo = table2.contents[3].contents[3].contents[0].
            replace(',','.')
        minimo = table2.contents[7].contents[3].contents[0].
            replace(',','.')
```

```
        else:
        prezzo = table1.contents[9].contents[3].contents[0].
            replace(',','.')
        variazione = table1.contents[11].contents[3].contents
            [0].replace(',','.')
        assoluta = table1.contents[13].contents[3].contents[0].
            replace(',','.')
        massimo = table2.contents[1].contents[3].contents[0].
            replace(',','.')
        minimo = table2.contents[7].contents[3].contents[0].
            replace(',','.')
```

Capitolo 6

Progettazione dell'interfaccia consultazione

6.1 Obiettivi

Dopo essere estratti ed immagazzinati nel database dal crawler, i dati sono pronti per essere analizzati. L'interfaccia grafica ha proprio il compito di permettere all'utente una veloce e pratica visualizzazione dei dati per permetterne l'analisi da parte dell'utente.

L'interfaccia grafica è sviluppata con il linguaggio Php; permette all'utente di inserire i titoli o i pacchetti da seguire, di visualizzare gli ultimi dati raccolti e di visualizzare, sotto forma di grafico, i dati storicizzati con una scala giornaliera, settimanale, mensile o annuale.

6.2 Funzioni dell'interfaccia

6.2.1 Funzioni di amministrazione

L'utente può aggiungere, modificare o elminare informazioni riguardanti un titolo. Nel caso dell'eliminazione, si è deciso che le informazioni rimarranno memorizzate nello storico, ma non verranno più effettuati gli aggiornamenti fino ad un'eventuale successiva riattivazione.

6.2.2 Funzioni di presentazione

L'utente può controllare l'andamento dei titoli immessi nel sistema attraverso delle tabelle di riepilogo e dei grafici. Questi hanno scale diverse per permettere

anche delle analisi a lungo termine e definire quindi l'andamento generale dei titoli. Inoltre, nel caso di un pacchetto, oltre ad essere visualizzata la tabella dell'asset allocation (inserita manualmente dall'utente), viene visualizzato un grafico a torta rappresentante i dati netti (azioni, obbligazioni, liquidità o altro) definiti nell'asset allocation.

6.3 Piattaforma di sviluppo

Per la realizzazione dell'interfaccia si è deciso di implementare un applicativo web-based. Questa soluzione offre diversi vantaggi quali:

- L'architettura Client-Server permette di gestire in maniera semplice più utenti
- L'interfaccia, costruita attraverso il browser, è facilmente apprendibile
- L'esecuzione, da parte dei client, necessita solo di un browser web disponibile su qualunque piattaforma.

6.3.1 Sistema operativo: ArchLinux

Figura 6.1: Logo di Arch Linux

Arch Linux è una distribuzione linux non specializzata, adattabile, progettata per soddisfare le necessità dell'utente linux competente. Usa pacchetti ottimizzati per architettura i686 e per questo funziona solo su processori di classe Pentium II o superiori. Arch Linux adopera un sistema a rolling release basato su due repository: Current e Release; il primo contiene tutti i pacchetti aggiornati all'ultima versione, mentre il secondo segue i rilasci (snapshot/ISO) semiregolari, e non viene aggiornato fino al successivo rilascio. Il repository Release è utile se si vuole aggiornare il sistema solo all'uscita di una nuova release.

La versione utilizzata per lo sviluppo e la fase di test del progetto è **Arch Linux - 2.6.34-ARCH**

6.3.2 Web server: Apache

Figura 6.2: Logo di Apache

Come server HTTP si è scelto di utilizzare Apache, rilasciato sotto l'omonima licenza opensource Apache License. Apache rappresenta lo standard de-facto dei web server: il grande successo di diffusione di questo software è l'indicatore più chiaro della qualità e dell'affidabilità di questo prodotto: secondo un'indagine Netcraft del 2005, su 75 milioni di siti web, circa 52 milioni utilizzavano Apache, ad ottobre 2006 il numero è salito a 60 milioni (69,32

La versione utilizzata per lo sviluppo e il test del progetto è **Apache - 2.2.15.**

6.3.3 Linguaggio di scripting: PHP

Figura 6.3: Logo di Php

PHP[1] è un linguaggio di scripting interpretato, con licenza open source e libera (ma incompatibile con la GPL), originariamente concepito per la programmazione Web ovvero la realizzazione di pagine web dinamiche. Attualmente è utilizzato principalmente per sviluppare applicazioni web lato server ma può essere usato anche per scrivere script a linea di comando o applicazioni standalone con interfaccia grafica. L'elaborazione di codice PHP sul server produce codice HTML da inviare al browser dell'utente che ne fa richiesta.

La versione utilizzata per l'implementazione del progetto è **PHP - 5.3.2-6**

[1] *Hypertext Preprocessor, preprocessore di ipertesti*

6.3.4 Librerie grafiche: JpGraph

Figura 6.4: Logo della libreria JpGraph

JpGraph è una libreria per la creazione di grafici OO[2] per le versioni di PHP superiori alla 5.1. La libreria è completamente scritta in PHP e pronta per essere utilizzata in qulunque script PHP. Questa libreria permette di creare con facilità grafici a linee, barre o a torta, ma anche grafici più complessi come quelli per la descrizione di impulsi, campi, polarità, intervalli e molti altri. E' disponibile anche una versione professionale che integra ulteriori tipologie di grafici.

La versione utilizzata per l'implementazione del progetto è **JpGraph - 3.0.7**

[2] *Object Oriented*

Capitolo 7

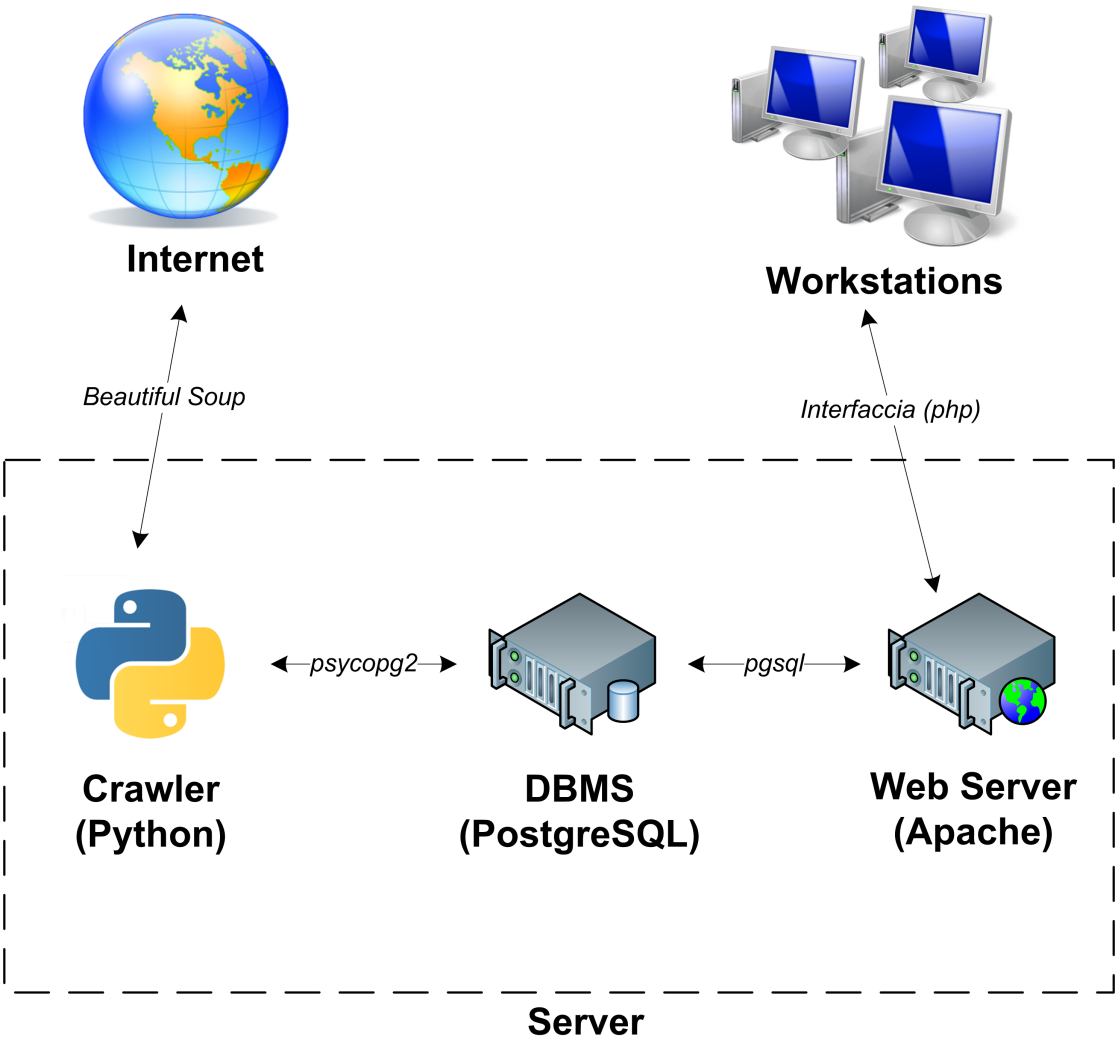
Implementazione

7.1 Obiettivi

Il sistema si compone di tre parti distinte che, mediante specifiche librerie, collaborano per ricercare, memorizzare ed analizzare i dati dei titoli scelti dall'utente. Ci sarà dunque un server centrale che ospiterà il database (PostgreSQL), il web server (Apache) per fornire l'interfaccia ai terminali, ed il crawler (Python) incaricato di ricercare i dati sul web.

Il crawler si interfaccia con il web attraverso la libreria *Urllib* per l'apertura delle pagine, mentre per la creazione del "parse-tree" viene utilizzata la libreria *Beautiful Soup*; infine, per l'inserimento dei dati nel database viene utilizzata la libreria *Psycopg2*. L'interfaccia, scritta in linguaggio *Php*, è resa disponibile dal web server Apache, e si interfaccia con il database utilizzando la libreria *PgSQL*.

L'architettura del sistema è riepilogata nella figura che segue.



7.2 Database: PostgreSQL

Per la scelta del DBMS da utilizzare, bisogna rifarsi alle tabelle esposte nel capitolo 2. Seguendo le valutazioni di queste tabelle, la scelta è caduta su PostgreSQL. Oltre ad essere un tipo di piattaforma open source, non ha limiti per quanto riguarda le dimensioni del database (vedi sezione 3.4) e presenta tutte le funzionalità necessarie alla gestione dei dati di interese. Per gestire al meglio i dati ed evitare inutili sprechi di spazio, sono state create tre entità e due relazioni:

Entità

- **Investimento:** Per ogni investimento si intendono memorizzare il codice, che lo identifica univocamente, il nome, il tipo, la rendita, la scadenza. E' inoltre possibile memorizzare il Rating e la quantità nel caso di titoli od obbligazioni.
- **Storico:** Per ogni tupla di questa entità si intendono memorizzare codice e data, che la identificano univocamente, il prezzo la variazione ed infine il minimo ed il massimo della giornata.
- **Giornaliero:** Tutti gli elementi di questa entità sono identificati dal codice e dall'ora. Inoltre vengono memorizzati anche il prezzo e la variazione ogni trenta minuti.
- **Asset Allocation:** Questa entità memorizza, in percentuale, le parti che formano un pacchetto definito dall'utente. Ogni tupla, identificata da un codice, descrive la divisione in azioni, obbligazioni, liquidità ed altro, di ogni pacchetto inserito dall'utente.

Relazioni

- **Storicizzazione:** Questa relazione collega l'entità *Investimento* con *Storico* e presenta un'unico attributo Data.
- **Composizione:** Questa relazione connette *Asset Allocation* ad *Investimento* e non presenta attributi.
- **Desctizione:** Questa relazione collega le entità *Storico* e *Giornaliero* e non presenta attributi

7.2.1 Progettazione concettuale

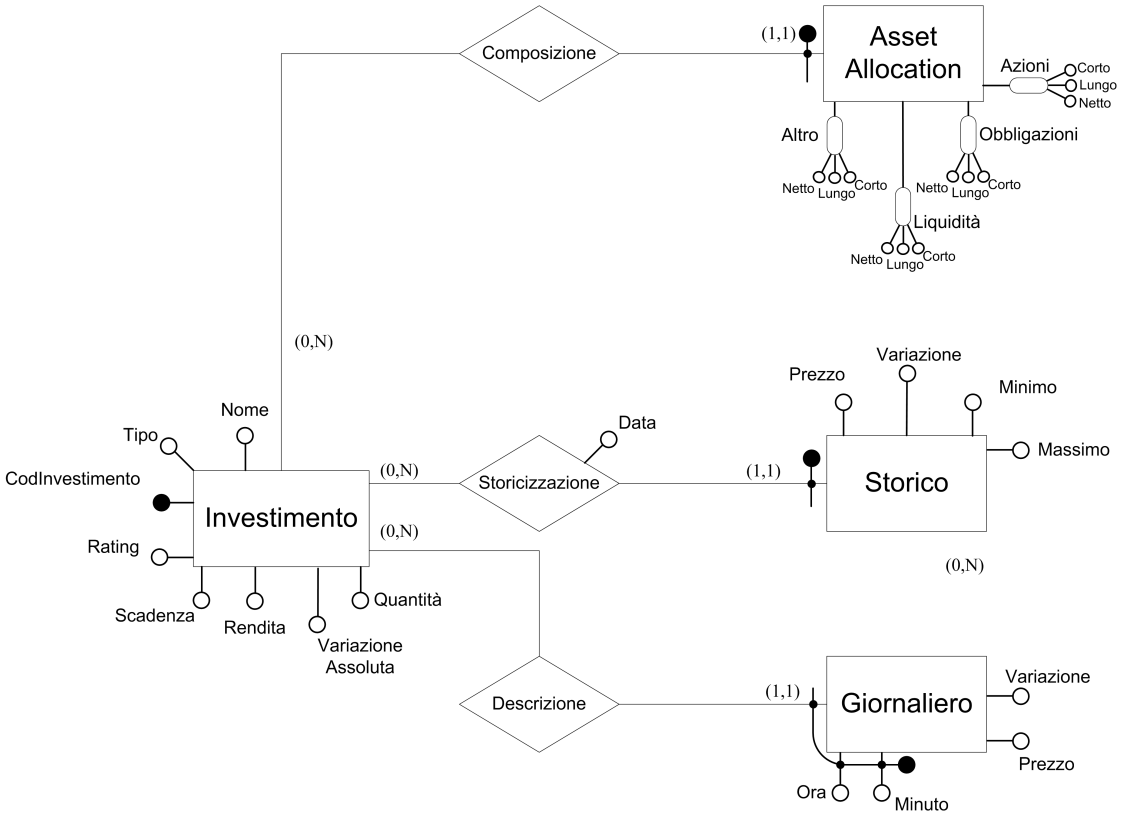


Figura 7.1: Schema ER

Regole di vincolo

Per completare la descrizione della realtà fornita dall'ER sono necessarie alcune regole di vincolo:

- Regola di vincolo 1: Una tupla con attributo tipo diverso da p non può avere un'Asset Allocation
- Regola di vincolo 2: Non è possibile memorizzare informazioni di un investimento dopo la data di scadenza

7.2.2 Progettazione logica

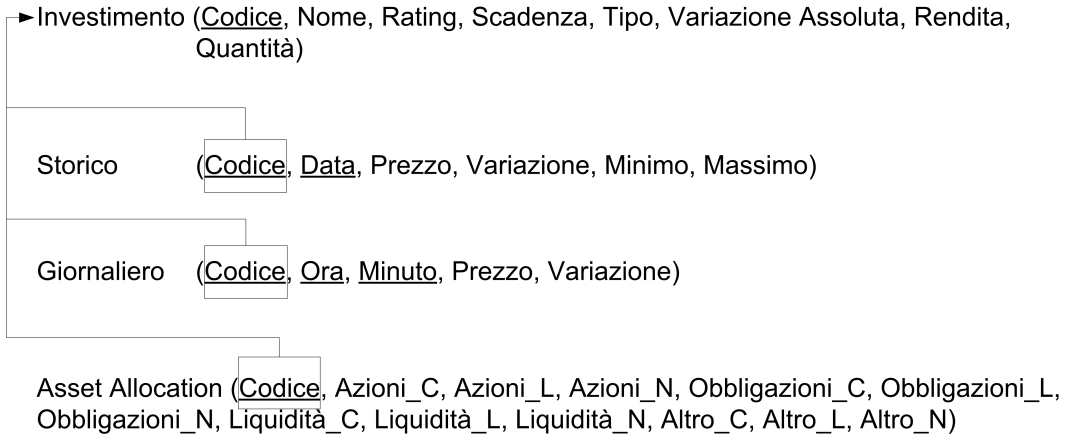


Figura 7.2: Schema logico

Schema logico, Regole di vincolo

- (RV1): Un investimento NON può avere quantità negativa
- (RV2): Gli attributi nome e tipo NON DEVONO essere nulli.
- (RV3): L'attributo ora NON DEVE essere inferiore a "9" o superiore a "17"

7.2.3 Codice SQL

```
CREATE TABLE investimento
(
  codice character varying(12) NOT NULL,
  nome character varying(60) NOT NULL,
  rating character varying(5),
  scadenza date,
  tipo character(1) NOT NULL,
  var_ass real,
  rendita real,
  quantita smallint NOT NULL DEFAULT 1,
  CONSTRAINT "investimento_PK" PRIMARY KEY (codice),
  CONSTRAINT "tipo_Check" CHECK (tipo = 'a'::bpchar OR tipo = 'o
      '::bpchar OR tipo = 'p'::bpchar OR tipo = 'i'::bpchar)
)
```

```
CREATE TABLE giornaliero
(
  codice character varying(12) NOT NULL,
  ora smallint NOT NULL,
  prezzo real NOT NULL,
  variazione real NOT NULL,
  minuto smallint NOT NULL,
  CONSTRAINT "giornaliero_PK" PRIMARY KEY (codice, ora, minuto),
  CONSTRAINT "giorna_invest_FK" FOREIGN KEY (codice)
      REFERENCES investimento (codice) MATCH SIMPLE
      ON UPDATE CASCADE ON DELETE CASCADE
)
```

```
CREATE TABLE storico
(
  codice character varying(12) NOT NULL,
  data date NOT NULL,
  prezzo real NOT NULL,
  varianza real NOT NULL,
  min real,
  max real,
  CONSTRAINT "storico_PK" PRIMARY KEY (codice, data),
  CONSTRAINT "storico_invest_FK" FOREIGN KEY (codice)
      REFERENCES investimento (codice) MATCH SIMPLE
      ON UPDATE CASCADE ON DELETE CASCADE
)
```

```
CREATE TABLE asset_allocation
(
  codice character varying(12) NOT NULL,
  azioni_c real,
  azioni_l real,
  azioni_n real,
  obbligazioni_c real,
  obbligazioni_l real,
  obbligazioni_n real,
  liquidita_c real,
  liquidita_l real,
  liquidita_n real,
  altro_c real,
  altro_l real,
  altro_n real,
  CONSTRAINT "assett_allocation_PK" PRIMARY KEY (codice),
```

```
  CONSTRAINT "asset_invest_FK" FOREIGN KEY (codice)
      REFERENCES investimento (codice) MATCH SIMPLE
      ON UPDATE CASCADE ON DELETE CASCADE
)
```

7.3 Librerie per l'interfacciamento con il database

Secondo quanto riportato nel capitolo 2, il DBMS scelto per l'implementazione del database è PostgreSQL: oltre ad essere gratuito, non impone limiti di crescita al database, e mette a disposizione tutti gli strumenti necessari alla storicizzazione e all'interrogazione dei dati.
Chiaramente, è necessario scegliere delle apposite librerie, che permettano al crawler di comunicare con la base di dati e, quindi, di poter inserire i dati estrappolati dai siti.

Software	Licenza	Piattaforma	Versione Python	DB API 2.0	Nativo (utilizza libpq)
Psycopg	LGPL	Unix, Win32	2.4-2.6	si	si
PyGreSQL	BSD	Unix, Win32	2.3-2.6	si	si
ocpgdb	BSD	Unix	2.3-2.6	si	si
py-postgresql	BSD	qualsiasi	3.0+	si	no
bpgsql	LGPL	qualsiasi	2.3-2.6	si	no
pg8000	BSD	qualsiasi	2.5+/3.0+	si	no

Tabella 7.1: Fonte: http://wiki.postgresql.org/wiki/Python

7.4 Psycopg

Psycopg è una libreria per l'interfacciamento ad un database PostgreSQL attraverso il linguaggio Python. I suoi punti di forza sono la capacità di supportare completamente le Python DB API 2.0 e la sicurezza dei suoi thread che possono condividere le connessioni. Venne creato per le applicazioni che usano in maniera pesante il multi-thread, creando e distruggendo molti cursori che compiono numerose operazioni di inserimento (INSERT) o modifica (UPDATE) sul database.

Pyscopg 2 è quasi una completa rivisitazione della prima versione. Le sue caratteristiche completano il protocollo libpq (v3), le funzioni COPY TO/COPY FROM e l'adattamento di tutte le classi base di dati di Python: stinghe (anche in unicode), interi, long, float, buffer (oggetti binari), boleani e i dati di tipo datetime. Inoltre supporta le query in unicode e le liste Python mappate in array PostgerSQL.

7.5 Crawler: Panoramica del codice

Come accennato nel capitolo 5, il crawler è composto principalmente da due parti: la prima si occupa dell'estrazione dei dati dalla pagina web mentre la seconda si occupa di inserire i dati nel database. Di seguito è esposto il codice per l'inserimento dei dati nel database.

```
conn = psycopg2.connect("dbname=Finanza user=crawler")
cur_modify = conn.cursor()
cur_modify.execute("SELECT codice FROM investimento WHERE codice
    ='"+isin+"';")
codice = cur_modify.fetchone()

if codice is None:
    cur_modify.execute("INSERT INTO investimento (codice, nome,
        tipo, var_ass) VALUES (%s, %s, %s, %s)", (isin, titolo,
        tipo, assoluta))
else:
    cur_modify.execute("UPDATE investimento SET var_ass= '"+
        assoluta+"' WHERE codice= '"+isin+"';")

cur_modify.execute("DELETE FROM storico WHERE codice='"+isin+"'
    AND data='"+str(today)+"' ;")
cur_modify.execute("INSERT INTO storico VALUES (%s, %s, %s, %s,
    %s, %s)", (isin, today, prezzo, variazione, massimo, minimo)
    )
```

```
cur_modify.execute("DELETE FROM giornaliero WHERE codice ='"+
    isin+"' AND ora ='"+str(hour)+"' AND minuto ='"+str(minute)+
    "' ;")
cur_modify.execute("INSERT INTO giornaliero VALUES (%s, %s, %s,
    %s, %s)", (isin, hour, prezzo, variazione, minute))

conn.commit()
cur_modify.close()
```

7.6 Interfaccia: Panoramica del codice

Anche per l'interfaccia grafica è necessario utilizzare specifiche librerire per effettuare operazione di lettura o scrittura su database. Php fornisce delle estensioni per l'utilizzo di diversi database, compreso PostgerSQL, non necessitando dunque di librerie esterne come per il crawler. Il codice che segue raccoglie degli esempi dell'utilizzo di queste funzioni, tratti da diverse pagine dell'interfaccia.

```
[ ... ]

$conn = pg_connect("dbname=Finanza user=web");
$menu = pg_query($conn, "SELECT codice, nome, tipo FROM
    investimento ORDER BY nome");

if(isset($_GET["graph"]))    $graph = $_GET["graph"];
else     $graph ="24 Ore";

echo <<<EOT
    <div align="center">
        <form name="input" action="db.php" method="get"/>
        <select name="isin"/>
EOT;
        while ($row = pg_fetch_row($menu)){
                if(isset($_GET["isin"]) and $_GET["isin"]==$row
                    [0])
                        echo "<option SELECTED value=\"$row[0]\"
                            > $row[1]</option>";
                else
                        echo "<option value=\"$row[0]\"> $row
                            [1]</option>";
        }
        echo <<<EOT
                </select>
```

```
                    <input type="hidden" value="$graph" name="graph"
                        />
                    <input type="submit" value="Visualizza" />
                    </form>
EOT;

if(isset($_GET["isin"])){
    $isin = $_GET["isin"];
    $type = pg_query($conn, "SELECT tipo FROM investimento WHERE
        codice='".$isin."';");
    $type = pg_fetch_row($type);
    $type = $type[0];
    $result1 = pg_query($conn, "SELECT * FROM investimento where
        codice='" . $isin . "';");

    $result2 = pg_query($conn, "SELECT * FROM storico where
        codice='" . $isin . "' ORDER BY data DESC;");
    $row = pg_fetch_row($result1);
    if($type=="p") $result2 = pg_query($conn, "SELECT * FROM
        storico where codice='ftse-mib' ORDER BY data DESC;");
    echo "<hr>";

[ ... ]

function add($isin, $nome, $type){
    $conn = pg_connect("dbname=Finanza user=web");
    $sql = "INSERT INTO investimento (codice, nome, tipo) VALUES
        ('".$isin."', '".$nome."', '".$type."');";
    pg_query($conn,$sql);
}

[ ... ]
```

Capitolo 8

Test

8.1 Obiettivi

Dopo aver completato sia il crawler che l'interfaccia grafica, è necessario verificare il funzionamento sia il funzionamento dei singoli componenti che dell'intero sistema. Inoltre, per permettere una breve dimostrazione, è necessario un popolamento minimo della base di dati.

8.2 Piattaforma per il collaudo

Per il collaudo del sistema è stata utilizzata la medesima piattaforma della fase di implementazione. Di seguito ne viene fornito un breve riepilogo.

8.2.1 Hardware

- Processore: Intel Pentium III @1GHz
- Memoria Ram: 256 MB @133Mhz
- Hard disk: 20GB

8.2.2 Software di base e d'ambiente

- Sistema operativo: **Arch Linux** - 2.6.34-ARCH
- Server grafico: **Xorg-server** 1.8.1.902-1
- Server web: **Apache** 2.2.15-2
- Interprete Php: **Php** 5.3.2-6

- Interprete Python: **Python** 2.6.5-3
- Librerie
  - Php: **JpGraph** 3.1.7p
  - Python: **Psycopg** 2.2.2

8.2.3 Software applicativo

- Editor di testo: **Gedit** 2.30.3-1
- Editor grafico: **Seashore** 0.5.1
- Browser web: **Chromium** 5.0.375.99-1
- Terminale: **Terminal** 0.4.5-1

8.3 Fasi e durate dei test

Le fasi dei test sono principalmente due:

- Test del **crawler**: durante questa fase si è verificato che il crawler riuscisse ad estrappolare con successo le informazioni dalle pagine web utilizzando le librerie preposte, e che l'inserimento di queste nel databse andasse a buon fine. Inoltre sono state fatte diverse prove per verificare il corretto funzionamento della funzione sleep(), nei diversi casi in cui viene richiamata.
- Test dell'**interfaccia**: durante questa fase si è verificato che i dati inseriti nel database dal crawler fossero accesibili, e che le librerie per la costruzione dei grafici funzionassero a dovere. Inoltre sono state testate tutte le funzioni di inserimento/cancellazione/modifica dell'interfaccia per controllare che vi fosse una corretta risposta da parte del database.

Chiaramente, durante tutto lo sviluppo delle componenti (crawler ed interfaccia) sono stati fatti dei test parziali per verificarne il funzionamento.

```
[luca@Portatile-Luca crawler]$ python crawler.py
Fiat
Prezzo: 10.85
Variazione: +0.00 %
Variazione assoluta: +0.00 %

Parmalat
Prezzo: 1.942
Variazione: +0.26 %
Variazione assoluta: +0.005 %

Pirelli & C
Prezzo: 5.96
Variazione: +0.68 %
Variazione assoluta: +0.04 %

Roma
Prezzo: 1.195
Variazione: +2.49 %
Variazione assoluta: +0.029 %

Arena
Prezzo: 0.0251
Variazione: -1.18 %
Variazione assoluta: -0.0003 %
Enel
Prezzo: 3.9675
Variazione: +0.63 %
Variazione assoluta: +0.025 %
Dormo secondi: 887
```

Figura 8.1: Test del crawler: acquisizione dati

```
   codice    |    data    |  prezzo  | varianza |  min   |  max
-------------+------------+----------+----------+--------+--------
 IT0000336518 | 2010-08-27 | 0.8790   |     0.11 |  0.881 |  0.856
 IT0003128367 | 2010-08-27 | 3.7400   |     1.22 |  3.743 |  3.665
 IT0001976403 | 2010-08-27 | 9.3500   |     0.97 |   9.35 |   9.09
 IT0003826473 | 2010-08-27 | 1.9190   |     1.05 |   1.92 |  1.873
 IT0004623051 | 2010-08-27 | 5.2750   |     1.05 |   5.29 |   5.14
 IT0001008876 | 2010-08-27 | 0.8880   |    -2.04 |  0.909 |  0.885
 IT0001347175 | 2010-08-27 | 0.0270   |     0.74 |  0.028 |  0.027
 IT0001137345 | 2010-08-27 | 9.1950   |     1.66 |  9.215 |   8.93
 IT0001279501 | 2010-08-27 | 2.9380   |     0.51 |  2.958 |   2.88
 IT0000336518 | 2010-08-28 | 0.8790   |     0.11 | 0.8805 | 0.8555
 IT0003128367 | 2010-08-28 | 3.7400   |     1.22 | 3.7425 |  3.665
 IT0001976403 | 2010-08-28 | 9.3500   |     0.97 |   9.35 |   9.09
 IT0003826473 | 2010-08-28 | 1.9190   |     1.05 |   1.92 |  1.873
 IT0004623051 | 2010-08-28 | 5.2750   |     1.05 |   5.29 |   5.14
 IT0001008876 | 2010-08-28 | 0.8880   |    -2.04 |  0.909 |  0.885
 IT0001347175 | 2010-08-28 | 0.0272   |     0.74 | 0.0278 | 0.0268
 IT0001137345 | 2010-08-28 | 9.1950   |     1.66 |  9.215 |   8.93
 IT0001279501 | 2010-08-28 | 2.9375   |     0.51 | 2.9575 |   2.88
 ftse-mib     | 2010-08-30 | 19699.66 |    -0.59 |        |
--Ancora--
```

Figura 8.2: Test del crawler: verifica inserimento dati

Queste due immagini rappresentano uno dei test riguardanti il crawler. La prima rappresenta l'output del crawler stampato sul terminale. La seconda invece rappresenta una query effettuata sul database per verificare che l'inserimento dei

dati raccolti dal crawler fosse andato a buon fine. Seguono le immagini dei test dell'interfaccia grafica.

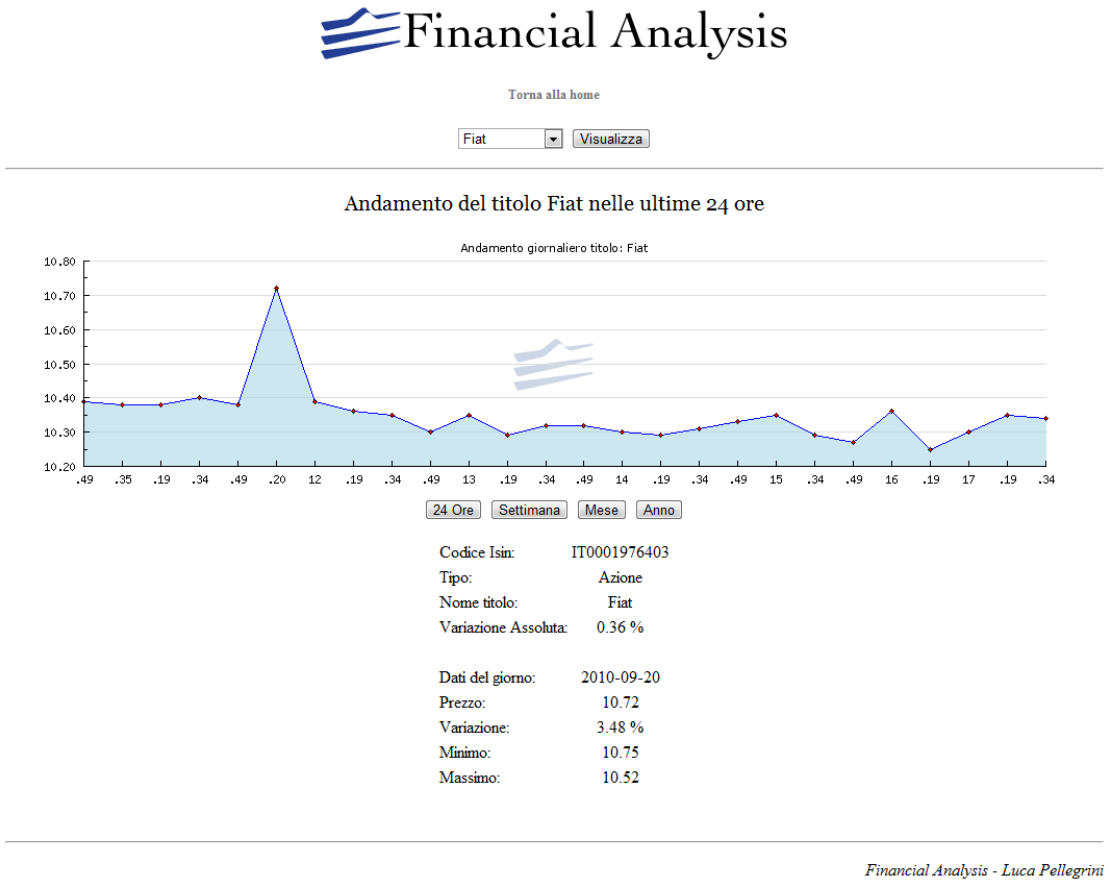


Figura 8.3: Test dell'interfaccia: analisi dei dati

Durante questo test si è verificato che i grafici venissero visualizzati correttamente. Nello specifico, qui viene testato il grafico con scala giornaliera. Seguono le immagini dei test della *Ricerca Titolo*.

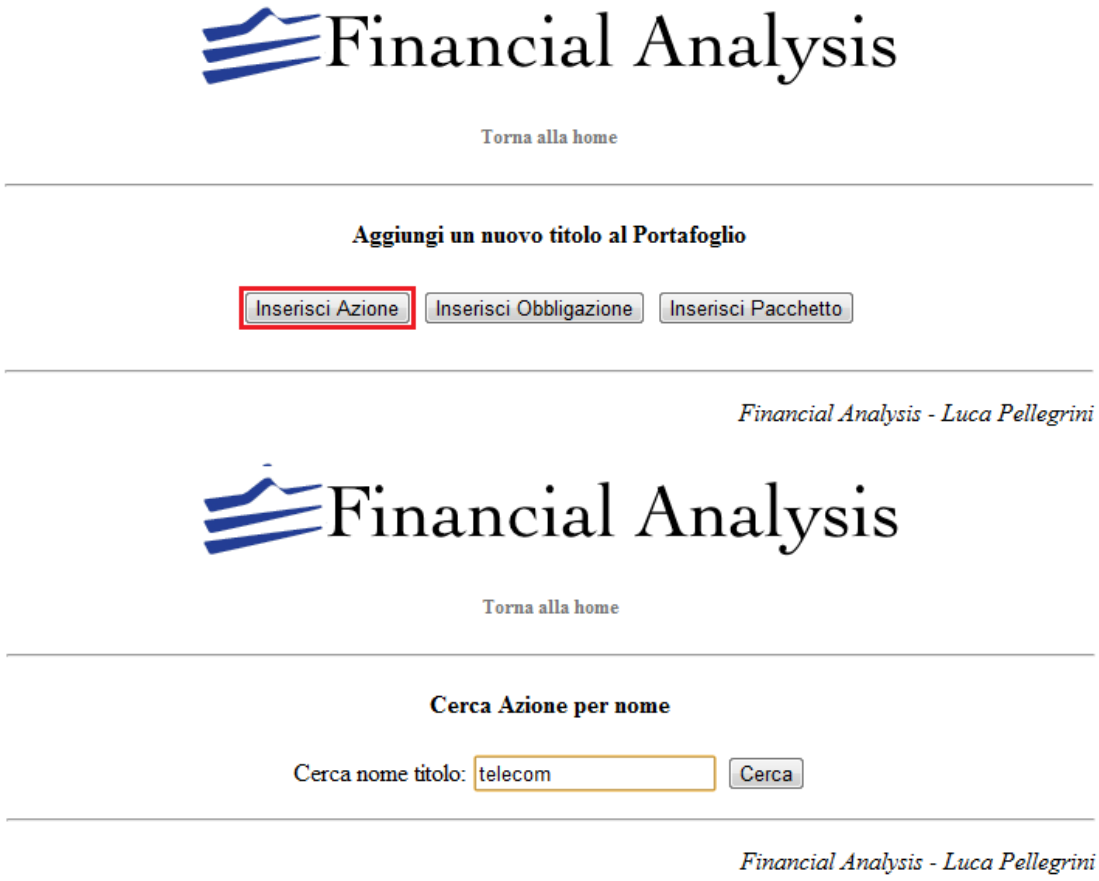


Figura 8.4: Test dell'interfaccia: ricerca titoli

Queste immagini rappresentano una delle sessioni di test nella quale si è verificato il corretto funzionamento della funzionalità di *Ricerca Titolo* dell'interfaccia. In questo caso specifico, è stata provata la ricerca di un titolo azionario *per nome (telecom)*. Di seguito l'immagine con i risultati.

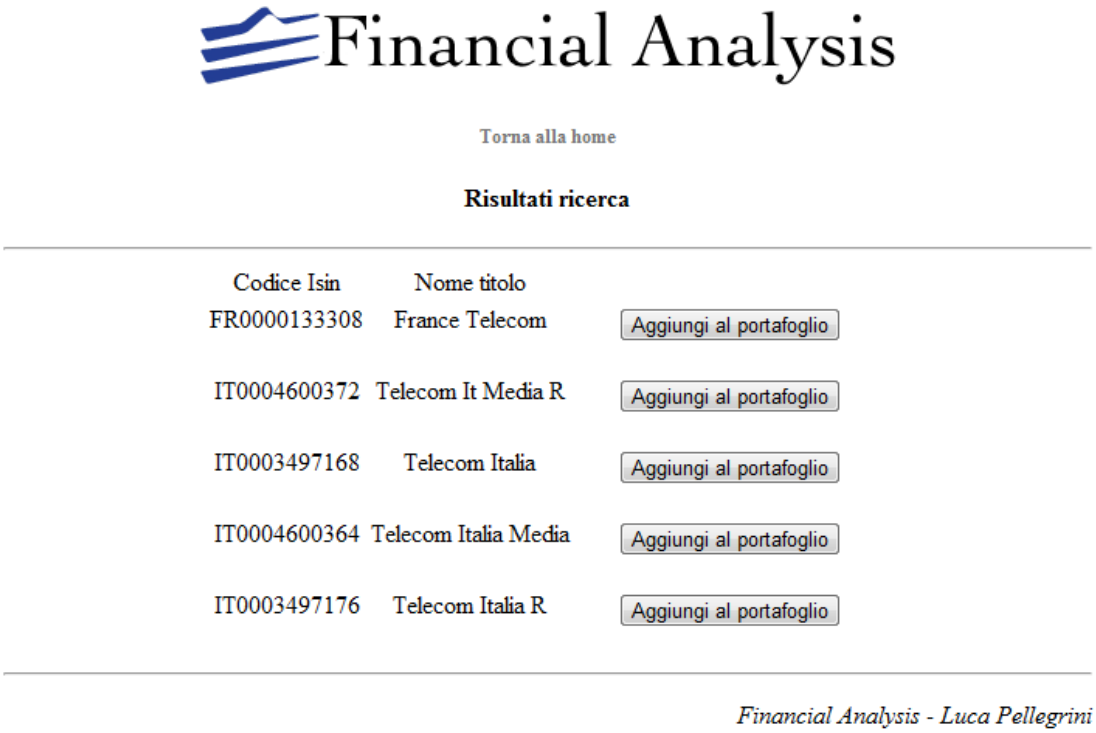


Figura 8.5: Test dell'interfaccia: visualizzazione risultati

Dopo aver completato crawler ed interfaccia, è stato testato il sistema creando le reali condizioni di utilizzo: una volta mandato in esecuzione, il crawler è autosufficiente e regola autonomamente gli intervalli per la raccolta dei dati; l'interfaccia è stata testata mediante l'utilizzo giornaliero verificando la corretta visualizzazione dei dati ed il corretto funzionamento di tutte le funzioni ausiliarie.

8.4 Conclusioni

Durante i test, sono emersi alcuni punti critici, sia per il crawler che per l'interfaccia.

- **Crawler**
  - È stato necessario gestire in maniera accorta il caso in cui non fosse disponibile la connessione al web.
  - Si sono verificati dei casi in cui, nonostante la connessione fosse attiva, i dati non fossero ancora disponibili.
  - Per gestire la finestra temporale di attività del crawler si è ricorsi alla funzione *sleep()*
- **Interfaccia**
  - Per la presentazione dei grafici, è necessario disporre di almeno due dati.
  - È stato necessario gestire l'eccezione creata dalla duplicazione di chiave primaria durante l'inserimento manuale dei titoli/pacchetti
  - Come per il crawler, si sono verificate difficoltà nel caso di connessione al web non disponibile (ricerca titoli)

Definiti dunque i punti critici, alla fine dei test, il risultato è comunque positivo. Il test è stato effettuato su una macchina prestazionalmente non eccellente proprio per verificare la disponibilità del sistema anche quando posto sotto stress (rete occupata, poca memoria disponibile). In conclusione il sistema non presenta particolari anomalie a appare sufficientemente stabile e pronto per l'utilizzo.

Bibliografia

[1] Sito ufficiale della Borsa Italiana, *http://www.borsaitaliana.it*

[2] Sito ufficiale de Il Sole 24 Ore, *http://www.ilsole24ore.com*

[3] Sito ufficiale di Milano Finanza, *http://www.milanofinanza.it*

[4] Sito ufficiale di PostgreSQL, *http://www.postgresql.org*

[5] Sito ufficiale di Arch Linux Italia, *http://www.archlinux.it*

[6] Sito ufficiale di Beautiful Soup, *http://www.crummy.com/software/BeautifulSoup*

[7] Sito ufficiale di Apache, *http://www.apache.org*

[8] Sito ufficiale di Php, *http://www.php.net*

[9] Sito ufficiale di JpGraph, *http://jpgraph.net*

[10] Sito ufficiale di Python, *http://www.python.org*

[11] AA.VV. (2008), *Appunti Universitari per Basi di Dati*, Padova

[12] Pilgrim, M., *Dive into Python, versione elettronica, http://it.diveintopython.org*