



---

INDIRIZZO IT - INFORMATICA E TELECOMUNICAZIONI

Esame di stato di istruzione secondaria

**SISTEMI PER LA  
GESTIONE DEI BIG DATA:  
IL CASO DI TRADING FOR YOU**

Elaborato di informatica e di sistemi e reti

**Tutor scolastico:**  
Prof. Roberto Picchiottino

**Presentato da:**  
André Ramolivaz

# Indice

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduzione</b>                               | <b>2</b>  |
| 1.1      | Quadro generale . . . . .                         | 2         |
| 1.2      | Presentazione del problema . . . . .              | 3         |
| 1.3      | Processo di sviluppo . . . . .                    | 4         |
| <b>2</b> | <b>Informatica</b>                                | <b>5</b>  |
| 2.1      | Strumenti utilizzati . . . . .                    | 5         |
| 2.2      | Raccolta ed analisi dei requisiti . . . . .       | 5         |
| 2.3      | Progettazione concettuale . . . . .               | 7         |
| 2.3.1    | Diagramma ER . . . . .                            | 7         |
| 2.4      | Progettazione logica . . . . .                    | 9         |
| 2.4.1    | Modello relazionale . . . . .                     | 9         |
| 2.4.2    | Specifiche delle entità . . . . .                 | 12        |
| 2.5      | Progettazione fisica . . . . .                    | 16        |
| 2.5.1    | Istruzioni SQL . . . . .                          | 16        |
| 2.6      | Accesso ai dati reali . . . . .                   | 21        |
| 2.6.1    | Richieste API . . . . .                           | 22        |
| 2.7      | Specifiche di sicurezza . . . . .                 | 23        |
| 2.7.1    | Cookies . . . . .                                 | 23        |
| 2.7.2    | Sessioni . . . . .                                | 24        |
| 2.7.3    | CSRF Token . . . . .                              | 24        |
| 2.8      | Form php per le operazioni . . . . .              | 25        |
| 2.9      | Specifiche delle operazioni . . . . .             | 27        |
| 2.9.1    | Visualizza operazioni del cliente x . . . . .     | 27        |
| 2.9.2    | Visualizza operazioni dalla più recente . . . . . | 27        |
| 2.9.3    | Filtro in base all'anno . . . . .                 | 28        |
| 2.9.4    | Filtro in base alla gestione . . . . .            | 28        |
| 2.9.5    | Altro . . . . .                                   | 28        |
| 2.10     | Sviluppo e descrizione . . . . .                  | 29        |
| <b>3</b> | <b>Sistemi e Reti</b>                             | <b>34</b> |
| 3.1      | Strumenti utilizzati . . . . .                    | 34        |
| 3.2      | Raccolta e analisi dei requisiti . . . . .        | 34        |
| 3.3      | Premessa . . . . .                                | 35        |
| 3.4      | Schema di rete (Sede Trading For You) . . . . .   | 35        |
| 3.4.1    | Descrizione generale . . . . .                    | 37        |
| 3.5      | Accesso al sistema per le operazioni . . . . .    | 38        |
| <b>4</b> | <b>Conclusioni</b>                                | <b>41</b> |
| 4.1      | SWOT Analysis . . . . .                           | 41        |

# Capitolo 1

## Introduzione

### 1.1 Quadro generale

*“I dati sono il petrolio del ventunesimo secolo e l’analisi è il motore a combustione” (Peter Sondergaard, Senior Vice President, Gartner)*

Negli ultimi anni si è sentito parlare sempre più spesso del trading online come una grande opportunità di ricavo immediato e cospicuo attraverso il semplice utilizzo del proprio PC di casa. Tecnicamente tutto ciò è possibile, poiché fare trading online significa comprare e vendere titoli o altri beni usando servizi di brokeraggio online, messi a disposizione da banche o da intermediari finanziari. Accedere ed operare facilmente sui mercati finanziari non garantisce, però, profitti perentorici e categorici: il trading è un’attività che comporta dei rischi e, pertanto, richiede l’adozione di opportune strategie basate principalmente su molteplici analisi tecniche.

Ecco che entra in gioco il ruolo delle tecniche avanza di gestione dei dati e machine learning. Queste fanno leva sull’utilizzo degli strumenti informatici in quanto facilitano notevolmente la gestione e analisi della mole di informazioni (in continua crescita) disponibili sui mercati finanziari e sulle società quotate (come ad esempio prezzi storici, volumi scambiati, analisi, news). Il grande limite riscontrato da anni consiste nell’impossibilità di analizzare manualmente ogni singolo dato per ricavarne indicatori chiari ed efficaci; elementi che possono essere d’aiuto nel processo decisionale (come un semplice grafico). La logica dei Data Base fornisce una soluzione a questa difficoltà, permette di organizzare, clusterizzare e armonizzare i dati in un formato che agevola le analisi permettendo l’automatizzazione di grafici ed indicatori. Lo si può immaginare come un riorganizzatore e sintetizzatore di informazioni.

In questo elaborato si propone una soluzione concreta al limite sopra descritto. Attraverso la creazione di una società fittizia denominata *Trading For You* che vede come obiettivo la gestione di un’enorme mole di dati, si crea una richiesta API per ottenere i dati e un’interfaccia che per permettere l’amministrazione di operazioni. Queste sono agevolate da uno studio dei dati che ha come output un grafico che incorpora i messaggi principali dell’andamento degli strumenti finanziari scelti.

I prossimi capitoli presentano una descrizione dei modelli di database utilizzati nel trading e una spiegazione dettagliata sullo specifico database oggetto di questo caso. Sempre in questo capitolo viene illustrata l’architettura del sistema creato unitamente ad una minuziosa descrizione delle funzionalità richieste dalla consegna dell’elaborato (ne sono un esempio i livelli di sicurezza e le query). Nel proseguo dell’elaborato, invece, verranno trattati gli argomenti inerenti l’architettura di rete necessaria alla sede (ipotetica) della *Trading For You*. In particolare verranno illustrati i migliori dispositivi, canali, protocolli e software necessari a costruire una struttura affidabile e sicura.

Infine, per concludere si presenterà una SWAT analisi sulla soluzione presentata e si proporranno delle idee di ulteriore sviluppo del progetto in merito l’opportunità di sfruttare algoritmi predittivi applicati al trading in un’ottica di ottimizzazione di portafoglio (gestione “auto”).

## 1.2 Presentazione del problema

In questa sezione si riporta la consegna che guida il lavoro e ne determina gli obiettivi.

I clienti che intendono usare la nuova piattaforma dovranno iscriversi (anche se già registrati a “My Bank” vista la scelta della società di rendere le piattaforme indipendenti). La TFY (*Trading For You*), controllata l’affidabilità del richiedente, gli fornisce le credenziali di accesso. La TFY quindi assegna ad ogni cliente un consulente. Il cliente può decidere la nazionalità del consulente tra i paesi dell’Unione Europea.

La piattaforma mette a disposizione molteplici indicatori di mercato relativi a differenti tipologie di prodotti di investimento: quali grafici e dati che aggiorna continuamente tramite richiesta RPC (o API) collegandosi all’apposito servizio delle borse di scambio (peer). In particolare per le azioni/FX (pre-selezionati) il sistema è in grado di fornire dei report applicando i vari principi di base (quali per esempio: oscillatore MACD 3/10, indicatore Rate of Change, indicatore ADX, Fibonacci, regole di Elliot) forniti dalle borse di scambio (pair) che costituiscono elementi utili al cliente ai fini decisionali e di pianificazione dei propri investimenti.

Ogni cliente ha la possibilità di impostare tramite le apposite opzioni disponibili sul sito web le seguenti procedure di gestione per ogni singola operazioni di acquisto o vendita (dei prodotti nel portfolio):

- “standard”: che richiede che il cliente convalidi manualmente ogni operazione;
- “auto” (automatica): che utilizza dei parametri soglia impostati dall’utente per l’avviare in automatico delle transazioni;
- “supported/assisted” (con supporto): che richiede la conferma da parte del cliente ad messaggio di suggerimento ricevuti dal proprio consulente.

Per ogni operazione si vogliono salvare: la data di esecuzione, il tipo (acquisto/vendita), la tipologia di gestione (automatica, standard e assistita) e l’importo in euro della transazione.

Il cliente deve poter accedere ai dati sia attraverso un PC che con lo smartphone. La sede della “Trading for you” occupa tutto un edificio su 4 piani nel quale lavorano circa 100 dipendenti, ognuno dei quali è dotato di una postazione fissa con PC e telefono. Inoltre, all’interno della sede, i dipendenti in base alle loro mansioni, devono poter accedere alle risorse tramite dispositivi mobili. Da tutti i dispositivi, deve essere possibile accedere a Internet.

I punti da sviluppare sono i seguenti:

1. Il progetto, anche mediante rappresentazioni grafiche, dell’infrastruttura tecnologica ed informatica necessaria a gestire il servizio nel suo complesso, dettagliando:
  - (a) l’architettura della rete presso la sede centrale descrivendo dispositivi, canali, protocolli e servizi software da implementare;
  - (b) l’architettura della rete mediante la quale i clienti potranno effettuare le operazioni utilizzando il proprio smartphone descrivendo dispositivi, canali, e protocolli;
  - (c) la soluzione migliore per il posizionamento del/dei server, motivando la scelta.
2. Il progetto della base di dati per la gestione del servizio sopra descritto: in particolare si richiedono il modello concettuale (E-R), il corrispondente modello logico e le istruzioni SQL che permettono di realizzarlo specificando anche le istruzioni che consentono di gestire l’integrità referenziale.
3. Il progetto e la codifica delle pagine web, eventualmente comprendente il logo, la pianificazione delle attività, le risorse umane utilizzate, relative ad almeno uno dei seguenti punti, che consentano al cliente autenticato:
  - (a) Di visualizzare l’elenco con le informazioni dettagliate delle operazioni di acquisto e di vendita relative all’anno ed indicare il saldo (somma vendite - somma acquisti);
  - (b) Come al punto precedente, ma l’utente può filtrare in base alla procedura di gestione dell’operazione (ovvero “standard”, “auto”, “assisted”).

(c) Ordinare la visualizzazione delle operazioni cronologicamente (dalla più recente).

Gestire opportunamente l'accesso autenticato alle eventuali aree riservate.

## 1.3 Processo di sviluppo

Per affrontare e completare il progetto, il lavoro è stato suddiviso in diverse attività organizzate nelle sei settimane disponibili come mostra il diagramma di Gantt sottostante.



Figura 1.1: Diagramma di Gantt

L'utilizzo del diagramma permette una ottimizzazione della pianificazione e coordinazione delle attività da svolgere. Queste ultime sono dettagliate in merito alla data d'inizio e di fine, ancillare ai vincoli di precedenza (ovvero il fatto che una task debba inevitabilmente precederne un'altra, a causa dell'indisponibilità della risorsa) e allo stato di avanzamento. Infine, il grafico dà una più accurata delle tempistiche che ogni step del progetto ha richiesto.

Le macro attività individuate, rappresentate con diversi colori nel diagramma, sono:

- **Analisi del problema**, ossia la lettura e comprensione dell'elaborato proposto;
- **Raccolta dati**, studio preliminare su come affrontare il problema e individualizzazione dei dati principali;
- **Progettazione Data base**: inizio della creazione del Data Base;
- **Configurazione ambiente di lavoro**: questa attività comprende XAMPP e l'utilizzo di XCode come strumenti per scrivere il codice e configurare il Data Base;
- **Codifica pagine web**, programmazione delle pagine Php e Html;
- **Progettazione della rete della sede**: ricerca degli strumenti necessari e simulazione della rete su packet tracer;
- **Elaborato di riepilogo**: stesura del seguente report in cui vengono illustrati dettagliatamente tutti i passi. Inoltre è prevista la presentazione per mostrare i risultati più significativi del lavoro svolto.

Nell'elaborato verranno approfonditi e dettagliatamente descritti tutti i vari step menzionati fino ad ora.

# Capitolo 2

## Informatica

L'incipit di questo capitolo è caratterizzato da un breve approfondimento sulle tecnologie e sugli strumenti utilizzati. Nelle altre sezioni, l'elaborato si dedica ai dati. Ne evidenzia l'utilità e l'applicabilità nell'ambito dell'analisi dei dati finanziari. Si conclude con un focus sull'autenticazione e sicurezza dell'applicativo e sulla creazione delle query principali, ne fa da cornice la descrizione delle varie funzionalità.

### 2.1 Strumenti utilizzati

Questa sezione è dedicata agli strumenti necessari, eccone un elenco:

- **Luchidchart**: un'applicazione web che permette di realizzare diversi tipi di diagrammi;
- **Apache**: il web server;
- **MYSQL**: un sistema per la gestione di basi di dati relazionali e quindi completare la progettazione fisica della base di dati;
- **PHP**: il linguaggio di programmazione interpretato per realizzare in modo veloce pagine dinamiche;
- **XCode**: un ambiente di sviluppo multi-linguaggio;
- Due **Web server**: insieme di risorse consumabili da un sistema software messe a disposizione nel caso proposto da questo elaborato sono [alphavantage](#) e [api.markestack](#).

### 2.2 Raccolta ed analisi dei requisiti

Procediamo in questa sezione alla raccolta ed analisi dei requisiti per cui si intende l'individuazione delle caratteristiche statiche (ovvero dei dati) e dinamiche (le cosiddette operazioni) dell'applicazione da realizzare.

Per iniziare, è importante individuare quale struttura di dati utilizzare per un software di trading. Da una scrematura di diversi tipi di sistemi di gestione di basi di dati, il migliore per il caso proposto è sicuramente il **modello relazionale**. Esso permette di raccogliere tutte le nostre informazioni in tabelle relazionate tra di esse per una facile rappresentazione logica e interrogazione dell'informazione. Le **tabelle** si basano a loro volta su degli **attributi** che saranno gli elementi in funzione dei dati che inseriremo. Sono gli stessi attributi la chiave delle relazioni intra-tabulari, le tabelle possono avere uno o più di questi in comune. Il loro ruolo è anche quello di riportare il **contenuto** del dato e la sua **tipologia** (i.e. BOOLEAN, VARCHAR, DATE, TIMESTAMP e INTEGER).

Considerando quanto appena descritto in merito al problema proposto, ci rendiamo conto che i dati da gestire sono relativi a:

- azione intesa come strumento finanziario, elemento che può essere scambiato sui mercati con azione di compra-vendita;

- utente che una volta iscritto ha la possibilità di aprire/chiedere le posizioni, visualizzare il saldo e lo storico delle transazioni;
- la scelta gestionale dell'utente a scelta tra: "standard", "auto" e "assisted".

Di conseguenza, le tabelle più significative ed utili da produrre sono:

1. *azioni*: composta dal simbolo, il nome dell'azienda e il mercato su cui è quotata ;
2. *cliente*: composta da nome cognome data di nascita email e password dell'utente ospite;
3. *conto*: caratterizzato dall'apposito codice e data di creazione;
4. *consulente*: dettagliato da utente della società "My Bank", identificato da nome cognome email e n° di telefono;
5. *portafoglio*: tabella che indicherà la tipologia di gestione del portafoglio, essa sarà composta quindi da nome e da data di creazione;
6. *portafoglio\_azioni*: entità che contiene le azioni in portafoglio, composto da quantità, tipologia e data di creazione;
7. *storico*: entità parallela alla precedente, composta dagli stessi attributi;
8. *valori\_giornalieri\_azione*: rappresenta i valori giornalieri della singola azione e quindi il prezzo di apertura, il prezzo massimo, il prezzo minimo, il prezzo chiusura, il volume e il giorno di scambio;
9. *valori\_cache\_azione*: tabella creata per comodità che estrapola i dati principali della singola azione, giorni di vita dell'azione, il prezzo alla sua nascita, il prezzo attuale e il massimo e minimo storico con le relative date.

Si identificano due tipologie di utenti: i "clienti" e i "consulenti" per cui si potrebbe inserire una gerarchia ISA di utente che abbia, come sotto entità, cliente e consulente. Analizzando meglio tale problematica però risulta migliore contraddistinguere le due entità in maniera distinta così da non creare generalizzazioni ed evitare problemi di interpretazione.

La creazione della tabella *valori\_cache\_values* è aggiuntiva. Questa permette di ottenere un piccolo report del titolo. In particolare questa risulta essere più efficace e rapida nel calcolo delle caratteristiche del valore giornaliero degli strumenti finanziari.

Le operazioni svolte nell'intero processo, e quindi, da gestire sono principalmente di quattro tipi, **inserimento**, **modifica** (o **eliminazione**) e **consultazione** dei dati:

- Operazioni di inserimento che comprendono:
  - Inserimento dei dati relativi al cliente e consulente;
  - Inserimento dei dati relativi all'azione;
  - Inserimento dei dati relativi alle posizioni aperte nel portafoglio, quindi degli stessi una volta chiusi, e nello storico.
- Operazioni di modifica o eliminazione:
  - Eliminazione dei dati delle posizioni aperte.
- Operazioni di consultazione di cui:
  - Consultazioni dei dati del cliente e del consulente;
  - Consultazione dei dati relativi a ogni singola azione (comprendendo grafici e report);
  - Consultazione dei dati relativi ai portafogli e alle relative posizioni aperte;
  - Consultazione dello storico transazioni.

Queste operazioni possono essere guidate e svolte con il linguaggio SQL. Si procede dalla creazione e rimozione del database stesso ad interventi architetturali (e.g. creazione, modifica e distruzione di tabelle e altri elementi interni del database). Importanti sono le elaborazioni di dati, trattati nei paragrafi seguenti.

Le operazioni fondamentali per la lettura e la manipolazione dei dati in SQL si contraddistinguono in quattro tipologie:

- *Create*: inserimento di nuovi dati;
- *Read*: lettura delle informazioni, le query menzionate in precedenza;
- *Update*: aggiornamento, modifica di dati già presenti nel database (aggiornamento dello storico e di correzioni sui mercati);
- *Delete*: cancellazione di dati presenti nel database.

L'acronimo che si forma con le iniziali dei termini "Create", "Read", "Update", "Delete" è **CRUD** ed è diventato il sinonimo – nei database ma anche in altri contesti – dell'insieme delle operazioni che possiamo svolgere su un set di dati.

## 2.3 Progettazione concettuale

Facendo leva sulle analisi sopra enunciate è opportuno inoltrarci nella fase di progettazione concettuale. Essa traduce le problematiche reali in uno schema concettuale facile da comprendere, senza focalizzarsi sulla struttura con cui viene costruito il database. Il modello Entità-Relazione (riportato da ora come ER) è solitamente usato durante questo tipo di progettazione per definire gli aspetti statici del sistema, cioè i dati. Esso descrive le entità da modellare, le relazioni che intercorrono tra di esse e le cardinalità delle relazioni. Con il termine cardinalità si intende il numero massimo e minimo di valori di un specifico attributo che possono essere associati a ciascuna istanza, nella Figura 2.1 è (1,N). Tra i vari attributi di una tabella molte volte è presente un attributo definito chiave. Questo ha la caratteristica di essere univoco e distinto in modo da poter identificare ogni tupla.

### 2.3.1 Diagramma ER

In base alle specifiche precedenti otterremo quindi il diagramma E-R sviluppato e allegato della pagina seguente, Figura 2.1.

**Chiave di lettura:** la cardinalità utilizzata per le relazioni è quella classica. Per esempio: un cliente è seguito da un solo consulente (rapporto 1 a 1) mentre i consulenti possono seguire un numero indefinito di clienti (ovvero da un minimo di 1 a un massimo di N clienti). Nel diagramma, la chiave primaria è evidenziata in grassetto e caratterizzata dal simbolo ▷;



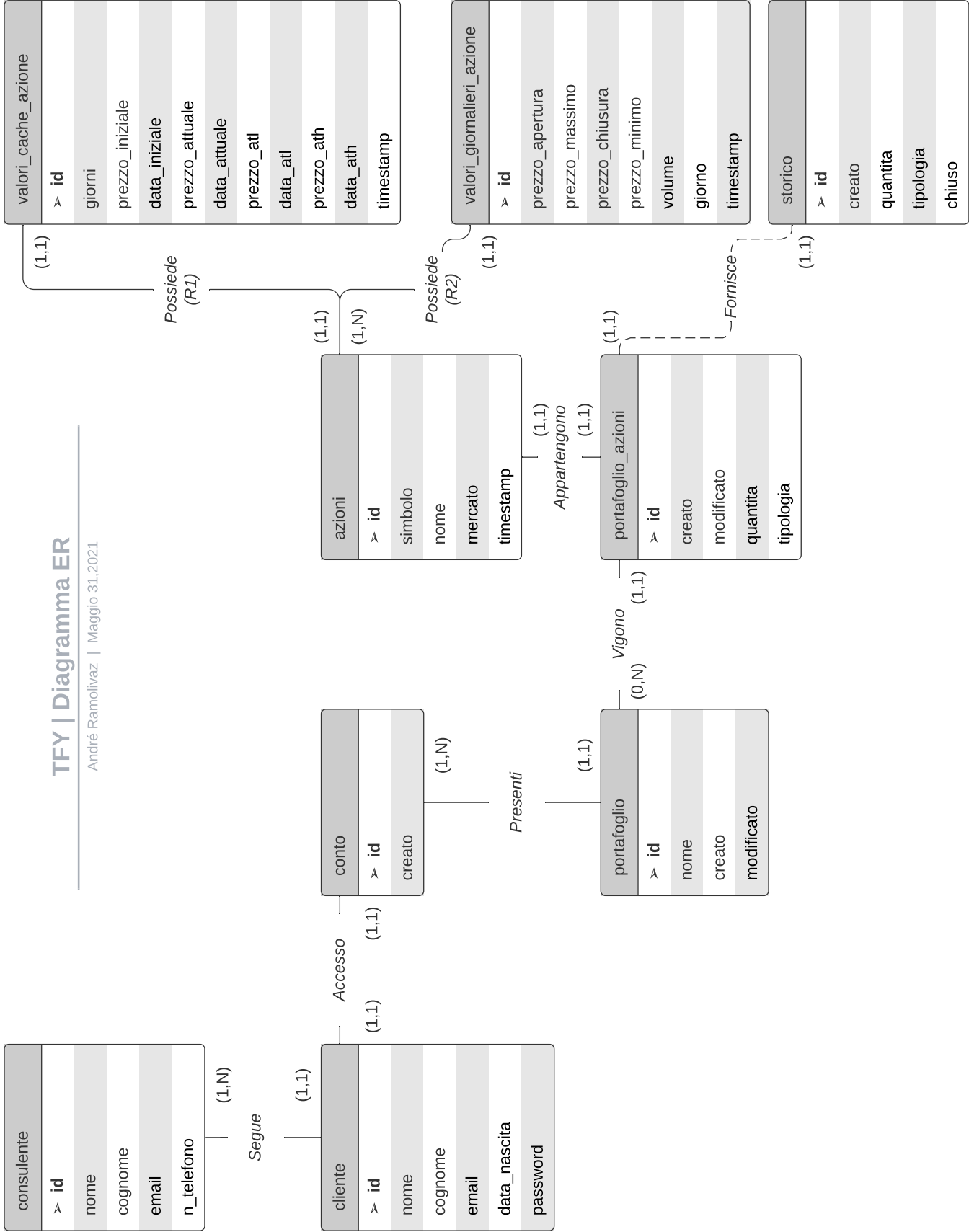


Figura 2.1: Progettazione concettuale

## 2.4 Progettazione logica

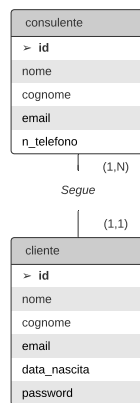
Infine, per concludere la fase di progettazione si creano più tabelle relazionali. Questo passaggio consiste nella traduzione dello schema concettuale eliminando i costrutti posti a più interpretazioni, seguendo specifiche regole. Queste si basano soprattutto sul tipo di relazione che lega due o più entità. Le tabelle infatti (come mostrato in Figura 2.1) possono essere collegate mediante 3 tipologie di relazione:

- **Uno a molti (1:N)**: una relazione dove ad ogni elemento si possono far corrispondere molti elementi di altre tabelle;
- **Molti a molti (N:N)**: una relazione dove più record possono essere associati a altrettanti record;
- **Uno a uno (1:1)**: relazione nella quale ciascun record è associato a un solo record di un'altra tabella.

Oltre al tipo di relazione in questa fase, risulta importante considerare altri aspetti tra cui: i vincoli di **dominio** (ogni attributo dev'essere in forma atomica), i vincoli di **integrità della chiave** (la chiave primaria non può mai avere un valore nullo), i vincoli di **integrità referenziale** (necessari al mantenimento della consistenza delle tuple) e per concludere i vincoli di **normalizzazione** (1FN, 2FN, 3FN). Tutti questi principi saranno messi in funzione su ciò che andremo a fare.

### 2.4.1 Modello relazionale

Tenendo a mente queste considerazioni inoltriamoci nella costruzione del modello relazionale. Analizzeremo step by step il modello Entità Relazione fino ad ottenimento del risultato finale. Ogni diagramma verrà spiegato nel paragrafo che lo segue a seguito della dicitura in grassetto **Interpretazione**.

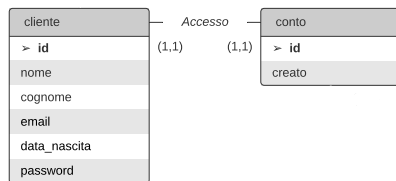


**Interpretazione:** un consulente può seguire al minimo 1 e al massimo N clienti, i quali, possono essere seguiti da un solo consulente.

Questa relazione **molti a uno** da origine a due schemi relazionali:

consulente(id, nome, cognome, email, n\_telefono)

cliente(id, nome, cognome, email, data\_nascita, password, idconsulente)

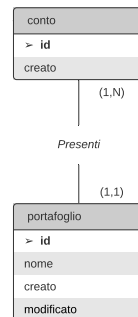


**Interpretazione:** un cliente può accedere al minimo 1 e al massimo 1 conto corrente, il quale, è accessibile da un solo cliente.

Questa relazione **uno a uno** da origine a due schemi relazionali:

cliente(id, nome, cognome, email, data\_nascita, password, idconsulente)

conto(id, creato, idcliente)

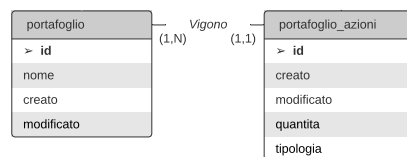


**Interpretazione:** in un conto vi possono essere attivi e al minimo 1 e al massimo N portafogli (es. Auto, Assisted, Standard) i quali, si rifanno ad un solo conto.

Questa relazione **uno a molti** da origine a due schemi relazionali:

conto(id, creato, idcliente)

portafoglio(id, nome, creato, modificato, idconto)

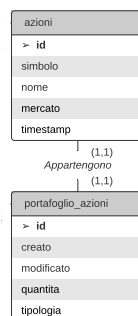


**Interpretazione:** in un portafoglio vi possono essere attive e al minimo 0 e al massimo N azioni, le quali si rifanno ad un solo portafoglio.

Questa relazione **uno a molti** da origine a due schemi relazionali:

portafoglio(id, nome, creato, modificato, idconto)

portafoglio.azioni(id, creato, modificato, quantita, tipologia, idportafoglio)

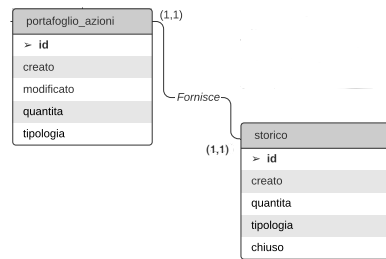


**Interpretazione:** l'azione di un portafoglio si riferisce ad al massimo un attributo della tabella azione e viceversa.

Questa relazione **uno a uno** da origine a due schemi relazionali:

portafoglio\_azioni(id, creato, modificato, quantita, tipologia, idportafoglio, idazione)

azioni(id, simbolo, nome, mercato, timestamp)

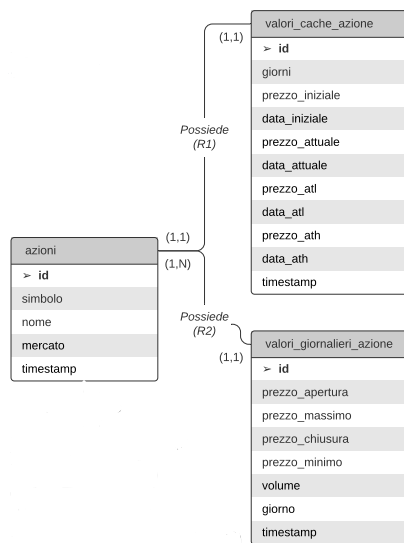


**Interpretazione:** le azioni di un portafoglio forniscono i dati per lo storico della transazione. Di conseguenza una posizione aperta è associabile a uno storico di transazione e viceversa. Lo storico altro non è che un duplicato della struttura della tabella portafoglio\_azioni. Nel momento in cui una posizione viene chiusa, tutti i suoi dati vengono spostati nella tabella storico dalla tabella portafoglio\_azione.

Questa relazione **uno a uno** da origine a due schemi relazionali:

portafoglio\_azioni(id, creato, modificato, quantita, tipologia, idportafoglio, idazione)

storico(id, creato, modificato, quantita, tipologia, idportafoglio, idazione, idportafoglio\_azione)



**Interpretazione:** le azioni ottengono i dati giornalieri N volte. Al tempo stesso, questi dati, possono essere associati ad una sola azione e non di più. Al contrario i valori cache rappresentano valori statici che non si aggiungono giornalmente. Per questo motivo si avrà una relazione 1 a 1 (un azione possiede una riga di valori cache e viceversa).

Queste relazioni **uno a uno e uno a molti** danno origine a tre schemi relazionali:

azioni(id, simbolo, nome, mercato, timestamp)

valori\_cache\_azione(id, giorni, prezzo\_iniziale, data\_iniziale, prezzo\_attuale, data\_attuale, prezzo\_atl, data\_atl, prezzo\_ath, data\_ath, timestamp, idazione)

valori\_giornalieri\_azione(id, prezzo\_apertura, prezzo\_massimo, prezzo\_chiusura, volume, giorno, timestamp, idazione)

## SINTETIZZANDO

Le tabelle relazionali finali saranno:

**consulente**(id, nome, cognome, email, n\_telefono)

**cliente**(id, nome, cognome, email, data\_nascita, password, idconsulente)

**conto**(id, creato, idcliente)

**azioni**(id, simbolo, nome, mercato, timestamp)

**valori\_cache\_azione**(id, giorni, prezzo\_iniziale, data\_iniziale, prezzo\_attuale, data\_attuale, prezzo\_atl, data\_atl, prezzo\_ath, data\_ath, timestamp, idazione)

**valori\_giornalieri\_azione**(id, prezzo\_apertura, prezzo\_massimo, prezzo\_chiusura, volume, giorno, timestamp, idazione)

**portafoglio**(id, nome, creato, modificato, idconto)

**portafoglio\_azioni**(id, creato, modificato, quantita, tipologia, idportafoglio, idazioni)

**storico**(id, creato, modificato, quantita, tipologia, idportafoglio, idazione, idportafoglio\_azione)

Le uniche tabelle che mostrano problematiche rilevanti sono: “portafoglio” e “conto”. “Portafoglio” è protagonista di una ridondanza nell’attributo della chiave esterna “idconto”. Questo non permette alla tabella di essere normalizzata e quindi garantire la massima qualità, efficacia e efficienza del nostro Data Base. Per porvi rimedio, si crea una tabella aggiuntiva riconosciuta con il nome di:

**p\_c**(id, idconto, idportafoglio)

Per ciò che riguarda le tabelle restanti, al contrario, esse rispettano le regole prima elencate e si presentano in:

**1FN** (Prima Forma Normale): tutti gli attributi sono atomici e sono identificati dalla chiave(Verificata);

**2FN** (Seconda Forma Normale): non ci sono dipendenze parziali dalla chiave (Verificata);

**3FN** (Terza Forma Normale): non sono presenti dipendenze transitive dalla chiave(Verificata).

## 2.4.2 Specifiche delle entità

Per completezza nel definire i dati, si inseriscono qui di seguito le specifiche di tutte le entità e le relative relazioni. Come si vedrà, viene definito non solo il tipo di attributo ma anche la sua associazione con le altre tabelle. Queste saranno le stesse che verranno inserite nella configurazione della base di dati su XAMPP.

| Colonna              | Tipo         | Null | Predefinito         | Collegamenti | Commenti                          |
|----------------------|--------------|------|---------------------|--------------|-----------------------------------|
| <i>id (Primaria)</i> | int(11)      | No   |                     |              | chiave auto incrementata          |
| simbolo              | varchar(255) | No   |                     |              | codice azione<br>es: GOOG         |
| nome                 | varchar(255) | No   |                     |              | nome società<br>quotata           |
| mercato              | varchar(255) | No   |                     |              | mercato di<br>scambio<br>es: NYSE |
| timestamp            | timestamp    | No   | current_timestamp() |              | data creazione                    |

Tabella 2.1: Azioni

| Chiave  | Tipo  | Unica | Compresso | Colonna | Cardinalità | Codifica | Null |
|---------|-------|-------|-----------|---------|-------------|----------|------|
| PRIMARY | BTREE | Sì    | No        | id      | 5           | A        | No   |
| symbol  | BTREE | Sì    | No        | simbolo | 5           | A        | No   |

Tabella 2.2: Indice-azioni

| Colonna                     | Tipo         | Null | Predefinito | Collegamenti             | Commenti                 |
|-----------------------------|--------------|------|-------------|--------------------------|--------------------------|
| <i>idcliente</i> (Primaria) | int(11)      | No   |             |                          | chiave auto incrementata |
| nome                        | varchar(255) | No   |             |                          | nome utente              |
| cognome                     | varchar(255) | No   |             |                          | cognome utente           |
| data_nascita                | varchar(255) | No   |             |                          | nascita utente           |
| cod_fiscale                 | varchar(255) | No   |             |                          | cod utente               |
| email                       | varchar(255) | No   |             |                          | email utente             |
| password                    | varchar(255) | No   |             |                          | psw utente               |
| idconsulente                | int(11)      | No   |             | consulente->idconsulente | Foreign Key              |

Tabella 2.3: Cliente

| Chiave       | Tipo  | Unica | Compresso | Colonna      | Cardinalità | Codifica | Null |
|--------------|-------|-------|-----------|--------------|-------------|----------|------|
| PRIMARY      | BTREE | Sì    | No        | idcliente    | 0           | A        | No   |
| idconsulente | BTREE | No    | No        | idconsulente | 0           | A        | No   |

Tabella 2.4: Indice-cliente

| Colonna                        | Tipo         | Null | Predefinito | Collegamenti | Commenti                 |
|--------------------------------|--------------|------|-------------|--------------|--------------------------|
| <i>idconsulente</i> (Primaria) | int(11)      | No   |             |              | chiave auto incrementata |
| nome                           | varchar(255) | No   |             |              | nome utente              |
| cognome                        | varchar(255) | No   |             |              | cognome utente           |
| email                          | varchar(255) | No   |             |              | email utente             |
| n_telefono                     | varchar(255) | No   |             |              | telefono utente          |

Tabella 2.5: Consulente

| Chiave  | Tipo  | Unica | Compresso | Colonna      | Cardinalità | Codifica | Null |
|---------|-------|-------|-----------|--------------|-------------|----------|------|
| PRIMARY | BTREE | Sì    | No        | idconsulente | 0           | A        | No   |

Tabella 2.6: Indice-consulente

| Colonna              | Tipo     | Null | Predefinito         | Collegamenti       | Commenti                 |
|----------------------|----------|------|---------------------|--------------------|--------------------------|
| <i>id</i> (Primaria) | int(11)  | No   |                     |                    | chiave auto incrementata |
| creato               | datetime | No   | current_timestamp() |                    | creazione                |
| idcliente            | int(11)  | No   |                     | cliente->idcliente | Foreign Key              |

Tabella 2.7: Conto

| Chiave    | Tipo  | Unica | Compresso | Colonna   | Cardinalità | Codifica | Null |
|-----------|-------|-------|-----------|-----------|-------------|----------|------|
| PRIMARY   | BTREE | Sì    | No        | id        | 0           | A        | No   |
| idcliente | BTREE | No    | No        | idcliente | 0           | A        | No   |

Tabella 2.8: Indice-conto

| Colonna              | Tipo    | Null | Predefinito | Collegamenti               | Commenti                 |
|----------------------|---------|------|-------------|----------------------------|--------------------------|
| <i>id (Primaria)</i> | int(11) | No   |             |                            | chiave auto incrementata |
| idconto              | int(11) | No   |             | conto->idconto             | Foreign Key              |
| idportafoglio        | int(11) | No   |             | portafoglio->idportafoglio | Foreign Key              |

Tabella 2.9: p.c

| Chiave        | Tipo  | Unica | Compresso | Colonna       | Cardinalità | Codifica | Null |
|---------------|-------|-------|-----------|---------------|-------------|----------|------|
| PRIMARY       | BTREE | Sì    | No        | id            | 0           | A        | No   |
| idconto       | BTREE | No    | No        | idconto       | 0           | A        | No   |
| idportafoglio | BTREE | No    | No        | idportafoglio | 0           | A        | No   |

Tabella 2.10: Indice-p.c

| Colonna              | Tipo         | Null | Predefinito         | Collegamenti | Commenti                 |
|----------------------|--------------|------|---------------------|--------------|--------------------------|
| <i>id (Primaria)</i> | int(11)      | No   |                     |              | chiave auto incrementata |
| nome                 | varchar(255) | No   |                     |              | nome gestione            |
| creato               | datetime     | No   | current_timestamp() |              | es: Standard creazione   |
| modificato           | datetime     | No   | current_timestamp() |              | modifica                 |

Tabella 2.11: Portafoglio

| Chiave  | Tipo  | Unica | Compresso | Colonna | Cardinalità | Codifica | Null |
|---------|-------|-------|-----------|---------|-------------|----------|------|
| PRIMARY | BTREE | Sì    | No        | id      | 3           | A        | No   |

Tabella 2.12: Indice-portafoglio

| Colonna              | Tipo         | Null | Predefinito         | Collegamenti               | Commenti                 |
|----------------------|--------------|------|---------------------|----------------------------|--------------------------|
| <i>id (Primaria)</i> | int(11)      | No   |                     |                            | chiave auto incrementata |
| idportafoglio        | int(11)      | No   |                     | portafoglio->idportafoglio | Foreign Key              |
| idazione             | int(11)      | No   |                     | azione->idazione           | Foreign Key              |
| creato               | date         | No   | current_timestamp() |                            | data creazione           |
| modificato           | timestamp    | No   | current_timestamp() |                            | data modifica            |
| quantita             | int(11)      | No   |                     |                            | n° azioni                |
| tipologia            | varchar(255) | No   |                     |                            | es: Compra e Vendi       |

Tabella 2.13: Portafoglio\_azioni

| Chiave  | Tipo  | Unica | Compresso | Colonna | Cardinalità | Codifica | Null |
|---------|-------|-------|-----------|---------|-------------|----------|------|
| PRIMARY | BTREE | Sì    | No        | id      | 7           | A        | No   |

Tabella 2.14: Indice-portafoglio\_azioni

| Colonna              | Tipo         | Null | Predefinito         | Collegamenti     | Commenti                             |
|----------------------|--------------|------|---------------------|------------------|--------------------------------------|
| <i>id (Primaria)</i> | int(11)      | No   |                     |                  | chiave auto incrementata             |
| giorni               | varchar(255) | No   |                     |                  | giorni dalla nascita della posizione |
| prezzo_iniziale      | varchar(255) | No   |                     |                  | prezzo inizio quotazione             |
| data_iniziale        | varchar(255) | No   |                     |                  | data inizio quotazione               |
| prezzo_attuale       | varchar(255) | No   |                     |                  | prezzo attuale                       |
| data_attuale         | varchar(255) | No   |                     |                  | data attuale                         |
| prezzo_atl           | varchar(255) | No   |                     |                  | massimo storico                      |
| data_atl             | varchar(255) | No   |                     |                  | data massimo                         |
| prezzo_ath           | varchar(255) | No   |                     |                  | minimo storico                       |
| data_ath             | varchar(255) | No   |                     |                  | data minimo                          |
| timestamp            | timestamp    | No   | current_timestamp() |                  | data auto                            |
| idazione             | int(11)      | No   |                     | azione->idazione | Foreign Key                          |

Tabella 2.15: valori\_cache\_azione

| Chiave  | Tipo  | Unica | Compresso | Colonna | Cardinalità | Codifica | Null |
|---------|-------|-------|-----------|---------|-------------|----------|------|
| PRIMARY | BTREE | Sì    | No        | id      | 318         | A        | No   |

Tabella 2.16: Indice-valori\_cache\_azioni

| Colonna              | Tipo         | Null | Predefinito         | Collegamenti     | Commenti                   |
|----------------------|--------------|------|---------------------|------------------|----------------------------|
| <i>id (Primaria)</i> | int(11)      | No   |                     |                  | chiave auto incrementata   |
| giorno               | date         | No   |                     |                  | giorno preso da web server |
| prezzo_apertura      | varchar(255) | No   |                     |                  | prezzo inizio quotazione   |
| prezzo_minimo        | varchar(255) | No   |                     |                  | giornaliera minimo         |
| prezzo_massimo       | varchar(255) | No   |                     |                  | giornaliero massimo        |
| prezzo_chiusura      | varchar(255) | No   |                     |                  | giornaliero prezzo azione  |
| volume               | varchar(255) | No   |                     |                  | chiusura giornaliera       |
| timestamp            | timestamp    | No   | current_timestamp() |                  | quantita                   |
| idazione             | int(11)      | No   |                     | azione->idazione | azioni scambiate           |
|                      |              |      |                     |                  | data auto                  |
|                      |              |      |                     |                  | Foreign Key                |

Tabella 2.17: valori\_giornalieri\_azione

| Chiave  | Tipo  | Unica | Compresso | Colonna | Cardinalità | Codifica | Null |
|---------|-------|-------|-----------|---------|-------------|----------|------|
| PRIMARY | BTREE | Sì    | No        | id      | 35021       | A        | No   |

Tabella 2.18: Indice-valori\_giornalieri\_azione



| Colonna              | Tipo         | Null | Predefinito         | Collegamenti               | Commenti   |
|----------------------|--------------|------|---------------------|----------------------------|--|
| <i>id (Primaria)</i> | int(11)      | No   |                     |                            | chiave auto incrementata                               |
| creato               | date         | No   |                     |                            | data apertura  |
| quantita             | int(11)      | No   |                     |                            | posizione<br>n°azioni                                  |
| chiuso               | date         | No   | current_timestamp() |                            | data chiusura<br>posizione<br>(apertura nello storico) |
| tipologia            | varchar(255) | No   |                     |                            | es: Compra, Vendi                                      |
| idazione             | int(11)      | No   |                     | azione->idazione           | Foreign Key  |
| idportafoglio        | int(11)      | No   |                     | portafoglio->idportafoglio | Foreign Key  |

Tabella 2.19: storico

| Chiave  | Tipo  | Unica | Compresso | Colonna | Cardinalità | Codifica | Null |
|---------|-------|-------|-----------|---------|-------------|----------|------|
| PRIMARY | BTREE | Sì    | No        | id      | 3           | A        | No   |

Tabella 2.20: Indice-storico

## 2.5 Progettazione fisica

La progettazione fisica presenta l'ultima stazione per completare questa prima parte di attività. In seguito, sono riportati in dettaglio i codici con la spiegazione dei rispettivi obiettivi.

### 2.5.1 Istruzioni SQL

Vista la descrizione a 360° del database, non resta che applicare le osservazioni fatte al software XAMPP. I costrutti per la creazione si basano sul linguaggio SQL (Structured Query Language). In allegato vediamo nel dettaglio i comandi che creano la struttura delle tabelle:

```

1  -- phpMyAdmin SQL Dump
2  -- version 5.1.0
3  -- https://www.phpmyadmin.net/
4  --
5  -- Host: localhost
6  -- Creato il: Mag 20, 2021 alle 18:00
7  -- Versione del server: 10.4.18-MariaDB
8  -- Versione PHP: 7.3.27
9  SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
10 START TRANSACTION;
11 SET time_zone = "+00:00";
12 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
13 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
14 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
15 /*!40101 SET NAMES utf8mb4 */;
16 --
17 -- Database: `DB_TradingForYou0`
18 --
19 --
20 --
21 -- Struttura della tabella `azioni`
22 --
23 -- Creazione: Mag 04, 2021 alle 10:00
24 --
25 CREATE TABLE `azioni` (
26   `id` int(11) NOT NULL,
27   `simbolo` varchar(255) NOT NULL,
28   `nome` varchar(255) NOT NULL,

```

```

29     `mercato` varchar(255) NOT NULL,
30     `timestamp` timestamp NOT NULL DEFAULT current_timestamp()
31 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
32 --
33 -- RELAZIONI PER TABELLA `azioni`:
34 --
35 -----
36 --
37 -- Struttura della tabella `cliente`
38 --
39 -- Creazione: Mag 20, 2021 alle 08:33
40 --
41 CREATE TABLE `cliente` (
42     `nome` varchar(255) NOT NULL,
43     `cognome` varchar(255) NOT NULL,
44     `data_nascita` date NOT NULL,
45     `cod_fiscale` varchar(255) NOT NULL,
46     `idcliente` int(11) NOT NULL,
47     `email` varchar(255) NOT NULL,
48     `password` varchar(255) NOT NULL,
49     `idconsulente` int(11) NOT NULL
50 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
51 --
52 -- RELAZIONI PER TABELLA `cliente`:
53 --     `idconsulente`
54 --         `consulente` -> `idconsulente`
55 --
56 -----
57 --
58 -- Struttura della tabella `consulente`
59 --
60 -- Creazione: Mag 04, 2021 alle 09:59
61 --
62 CREATE TABLE `consulente` (
63     `nome` varchar(255) NOT NULL,
64     `cognome` varchar(255) NOT NULL,
65     `email` varchar(255) NOT NULL,
66     `n_telefono` varchar(255) NOT NULL,
67     `idconsulente` int(11) NOT NULL
68 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
69 --
70 -- RELAZIONI PER TABELLA `consulente`:
71 --
72 -----
73 --
74 -- Struttura della tabella `conto`
75 --
76 -- Creazione: Mag 20, 2021 alle 08:28
77 --
78 CREATE TABLE `conto` (
79     `id` int(11) NOT NULL,
80     `creato` datetime NOT NULL DEFAULT current_timestamp(),
81     `idcliente` int(11) NOT NULL
82 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
83 --
84 -- RELAZIONI PER TABELLA `conto`:
85 --     `idcliente`
86 --         `cliente` -> `idcliente`
87 --
88 -----
89 --
90 -- Struttura della tabella `portafoglio`
91 --
92 -- Creazione: Mag 04, 2021 alle 10:07
93 --
94 CREATE TABLE `portafoglio` (
95     `id` int(11) NOT NULL,
96     `nome` varchar(255) NOT NULL,
97     `creato` timestamp NOT NULL DEFAULT current_timestamp(),
98     `modificato` timestamp NOT NULL DEFAULT current_timestamp()
99 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

100 --
101 -- RELAZIONI PER TABELLA `portafoglio`:
102 --
103 -- -----
104 --
105 -- Struttura della tabella `portafoglio_azioni`
106 --
107 -- Creazione: Mag 11, 2021 alle 08:31
108 -- Ultimo aggiornamento: Mag 20, 2021 alle 15:58
109 --
110 CREATE TABLE `portafoglio_azioni` (
111   `id` int(11) NOT NULL,
112   `idportafoglio` int(11) NOT NULL,
113   `idazione` int(11) NOT NULL,
114   `creato` date NOT NULL DEFAULT current_timestamp(),
115   `modificato` timestamp NOT NULL DEFAULT current_timestamp(),
116   `quantita` int(11) NOT NULL,
117   `tipologia` varchar(255) NOT NULL
118 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
119 --
120 -- RELAZIONI PER TABELLA `portafoglio_azioni`:
121 --
122 -- -----
123 --
124 -- Struttura della tabella `p_c`
125 --
126 -- Creazione: Mag 20, 2021 alle 08:31
127 --
128 CREATE TABLE `p_c` (
129   `idportafoglio` int(11) NOT NULL,
130   `idconto` int(11) NOT NULL,
131   `id` int(11) NOT NULL
132 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
133 --
134 -- RELAZIONI PER TABELLA `p_c`:
135 --   `idconto`
136 --   `conto` -> `id`
137 --   `idportafoglio`
138 --   `portafoglio` -> `id`
139 --
140 -- -----
141 --
142 -- Struttura della tabella `storico`
143 --
144 -- Creazione: Mag 11, 2021 alle 08:35
145 --
146 CREATE TABLE `storico` (
147   `id` int(11) NOT NULL,
148   `creato` date NOT NULL,
149   `quantita` int(11) NOT NULL,
150   `chiuso` date NOT NULL DEFAULT current_timestamp(),
151   `idportafoglio` int(11) NOT NULL,
152   `idazione` int(11) NOT NULL,
153   `tipologia` varchar(255) NOT NULL
154 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
155 --
156 -- RELAZIONI PER TABELLA `storico`:
157 --
158 -- -----
159 --
160 -- Struttura della tabella `valori_cache_azione`
161 --
162 -- Creazione: Mag 04, 2021 alle 10:03
163 --
164 CREATE TABLE `valori_cache_azione` (
165   `id` int(11) NOT NULL,
166   `idazione` int(11) NOT NULL,
167   `giorni` varchar(255) NOT NULL,
168   `prezzo_iniziale` varchar(255) NOT NULL,
169   `data_iniziale` varchar(255) NOT NULL,
170   `prezzo_attuale` varchar(255) NOT NULL,

```

```

171 `data_attuale` varchar(255) NOT NULL,
172 `prezzo_atl` varchar(255) NOT NULL,
173 `data_atl` varchar(255) NOT NULL,
174 `prezzo_ath` varchar(255) NOT NULL,
175 `data_ath` varchar(255) NOT NULL,
176 `timestamp` timestamp NOT NULL DEFAULT current_timestamp()
177 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
178 --
179 -- RELAZIONI PER TABELLA `valori_cache_azione`:
180 --
181 -- -----
182 --
183 -- Struttura della tabella `valori_giornalieri_azione`
184 --
185 -- Creazione: Mag 04, 2021 alle 10:06
186 --
187 CREATE TABLE `valori_giornalieri_azione` (
188   `id` int(11) NOT NULL,
189   `idazione` int(11) NOT NULL,
190   `prezzo_apertura` varchar(255) NOT NULL,
191   `prezzo_massimo` varchar(255) NOT NULL,
192   `prezzo_minimo` varchar(255) NOT NULL,
193   `prezzo_chiusura` varchar(255) NOT NULL,
194   `volume` varchar(255) NOT NULL,
195   `giorno` date NOT NULL,
196   `timestamp` timestamp NOT NULL DEFAULT current_timestamp()
197 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
198 --
199 -- RELAZIONI PER TABELLA `valori_giornalieri_azione`:
200 --
201 --
202 -- Indici per le tabelle scaricate
203 --
204 --
205 -- Indici per le tabelle `azioni`
206 --
207 ALTER TABLE `azioni`
208   ADD PRIMARY KEY (`id`),
209   ADD UNIQUE KEY `symbol` (`simbolo`);
210 --
211 -- Indici per le tabelle `cliente`
212 --
213 ALTER TABLE `cliente`
214   ADD PRIMARY KEY (`idcliente`),
215   ADD KEY `idconsulente` (`idconsulente`);
216 --
217 -- Indici per le tabelle `consulente`
218 --
219 ALTER TABLE `consulente`
220   ADD PRIMARY KEY (`idconsulente`);
221 --
222 -- Indici per le tabelle `conto`
223 --
224 ALTER TABLE `conto`
225   ADD PRIMARY KEY (`id`),
226   ADD KEY `idcliente` (`idcliente`);
227 --
228 -- Indici per le tabelle `portafoglio`
229 --
230 ALTER TABLE `portafoglio`
231   ADD PRIMARY KEY (`id`);
232 --
233 -- Indici per le tabelle `portafoglio_azioni`
234 --
235 ALTER TABLE `portafoglio_azioni`
236   ADD PRIMARY KEY (`id`);
237 --
238 -- Indici per le tabelle `p_c`
239 --
240 ALTER TABLE `p_c`
241   ADD PRIMARY KEY (`id`),

```

```

242     ADD KEY `idconto` (`idconto`),
243     ADD KEY `idportafoglio` (`idportafoglio`);
244 --
245 -- Indici per le tabelle `storico`
246 --
247 ALTER TABLE `storico`
248     ADD PRIMARY KEY (`id`);
249 --
250 -- Indici per le tabelle `valori_cache_azione`
251 --
252 ALTER TABLE `valori_cache_azione`
253     ADD PRIMARY KEY (`id`);
254 --
255 -- Indici per le tabelle `valori_giornalieri_azione`
256 --
257 ALTER TABLE `valori_giornalieri_azione`
258     ADD PRIMARY KEY (`id`);
259 --
260 -- AUTO_INCREMENT per le tabelle scaricate
261 --
262 --
263 -- AUTO_INCREMENT per la tabella `azioni`
264 --
265 ALTER TABLE `azioni`
266     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
267 --
268 -- AUTO_INCREMENT per la tabella `cliente`
269 --
270 ALTER TABLE `cliente`
271     MODIFY `idcliente` int(11) NOT NULL AUTO_INCREMENT;
272 --
273 -- AUTO_INCREMENT per la tabella `consulente`
274 --
275 ALTER TABLE `consulente`
276     MODIFY `idconsulente` int(11) NOT NULL AUTO_INCREMENT;
277 --
278 -- AUTO_INCREMENT per la tabella `conto`
279 --
280 ALTER TABLE `conto`
281     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
282 --
283 -- AUTO_INCREMENT per la tabella `portafoglio`
284 --
285 ALTER TABLE `portafoglio`
286     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
287 --
288 -- AUTO_INCREMENT per la tabella `portafoglio_azioni`
289 --
290 ALTER TABLE `portafoglio_azioni`
291     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
292 --
293 -- AUTO_INCREMENT per la tabella `p_c`
294 --
295 ALTER TABLE `p_c`
296     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
297 --
298 -- AUTO_INCREMENT per la tabella `storico`
299 --
300 ALTER TABLE `storico`
301     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
302 --
303 -- AUTO_INCREMENT per la tabella `valori_cache_azione`
304 --
305 ALTER TABLE `valori_cache_azione`
306     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
307 --
308 -- AUTO_INCREMENT per la tabella `valori_giornalieri_azione`
309 --
310 ALTER TABLE `valori_giornalieri_azione`
311     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
312 --

```

```

313 -- Limiti per le tabelle scaricate
314 --
315 --
316 -- Limiti per la tabella `cliente`
317 --
318 ALTER TABLE `cliente`
319     ADD CONSTRAINT `cliente_ibfk_1` FOREIGN KEY (`idconsulente`) REFERENCES `consulente`
        (`idconsulente`) ON DELETE CASCADE ON UPDATE CASCADE;
320 --
321 -- Limiti per la tabella `conto`
322 --
323 ALTER TABLE `conto`
324     ADD CONSTRAINT `conto_ibfk_1` FOREIGN KEY (`idcliente`) REFERENCES `cliente`
        (`idcliente`) ON DELETE CASCADE ON UPDATE CASCADE;
325 --
326 -- Limiti per la tabella `p_c`
327 --
328 ALTER TABLE `p_c`
329     ADD CONSTRAINT `p_c_ibfk_1` FOREIGN KEY (`idconto`) REFERENCES `conto` (`id`) ON
        DELETE CASCADE ON UPDATE CASCADE,
330     ADD CONSTRAINT `p_c_ibfk_2` FOREIGN KEY (`idportafoglio`) REFERENCES `portafoglio`
        (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
331 COMMIT;
332 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
333 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
334 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

## 2.6 Accesso ai dati reali

Avendo ultimato il database, è opportuno dedicare una sezione alla descrizione della metodologia di inserimento dei dati esterni. Nel caso in oggetto, desideriamo importare delle informazioni storiche e prettamente quantitative sugli strumenti azionari. Per le azioni scelte vengono aggiunte nelle tabelle denominate come “azioni” le variabili *valori\_giornalieri\_azione* e *valori\_cache\_azioni*. Il trasferimento di questi dati avviene con una richiesta API (Application Programming Interface).

Le richieste API sono una delle più importanti componenti strategiche della trasformazione digitale in quanto semplifica il dialogo tra fonti esterne e l'utilizzo che l'utente ne fa o richiede evitando ridondanze e automatizzando il processo. In altri termini, le richieste API si rifanno alle risorse e metodologie non destinate alla visualizzazione da parte dell'utente (l'user desidera ricevere le informazioni pronte ed organizzate, non è interessato a capire come vengono ottenute o costruite le tabelle). Queste sono di uso esclusivo da parte di applicazioni software che ricostruiscono e preparano il formato adatto alla lettura ed interpretazione dell'utente. L'insieme di queste risorse viene definito “web-service.” Per accedere a quest'ultimi occorre una chiave chiamata *API Key* (rilasciata a titolo gratuito) e conoscere le web API (ovvero le specifiche di composizioni degli URL consultabili dal web-service stesso). Una volta ottenute queste indicazioni è possibile accedere ai corrispettivi dati messi a disposizione.

Per lo sviluppo di questa fase si ritiene più idoneo l'utilizzo della tecnica ibrida di mashup. Essa permette la creazione di nuovi siti mescolando elementi da servizi diversi ovvero includendo dinamicamente informazioni o contenuti provenienti da più fonti. Questo ci consente di far leva sull'enorme quantità di dati presenti nel web e presentarli/collegarli per creare un servizio completamente nuovo.

I servizi che vengono utilizzati per soddisfare gli obiettivi stabiliti sono i seguenti:

- <https://www.alphavantage.co>: utilizzato per ottenere i dati giornalieri nelle diverse sfaccettature (e.g. massimo e minimo);
- <http://api.marketstack.com>: da cui si ricava il nome e altre informazioni qualitative sull'azione e sul suo sottostante.

Con questa procedura, i dati vengono stampati in JSON (JavaScript Object Notation), un formato che prevede l'utilizzo di elenchi gerarchici di liste di coppie chiave/valore che risultano immediatamente utilizzabili nella forma di array associativi.

## 2.6.1 Richieste API

Più nello specifico, per richiedere i dati al web service, l'utente deve specificare i parametri della richiesta nell'URL dopo il simbolo "?" in una lista denominata query string che utilizza come separatore il simbolo "&" e rispetti i rigidi criteri imposti dalla formulazione degli URL. Le macro richieste API saranno principalmente due.

### alphavantage

La prima richiesta si riferisce a alphavantage ed è un URL della forma:

```
https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=$simbolo&apikey=$key
```

Dove "\$simbolo" rappresenta la variabile del simbolo dell'azione che inserisce l'utente (per esempio si inserirà GOOG per selezionare l'azione di Alphabet - proprietaria di Google) e "\$key" è la API key ottenuta iscrivendosi al web-service (es: CKD2EM1LOXO15WS). Per chiarire ulteriormente questo esempio riportiamo in seguito il risultato ottenuto con il formato json:

```
1 {
2   "Meta Data": {
3     "1. Information": "Daily Prices (open, high, low, close) and Volumes",
4     "2. Symbol": "GOOG",
5     "3. Last Refreshed": "2021-05-20",
6     "4. Output Size": "Compact",
7     "5. Time Zone": "US/Eastern"
8   },
9   "Time Series (Daily)": {
10     "2021-05-20": {
11       "1. open": "2328.0400",
12       "2. high": "2360.3400",
13       "3. low": "2321.0900",
14       "4. close": "2356.0900",
15       "5. volume": "1191593"
16     },
17     "2021-05-19": {
18       "1. open": "2264.4000",
19       "2. high": "2316.7600",
20       "3. low": "2263.5200",
21       "4. close": "2308.7100",
22       "5. volume": "967544"
23     },
24     "2021-05-18": {
25       "1. open": "2336.9056",
26       "2. high": "2343.1500",
27       "3. low": "2303.1600",
28       "4. close": "2303.4300",
29       "5. volume": "865119"
30     }
31     [...]
32   }
33 }
```

Il codice mostra molto chiaramente i dati ottenuti ("volume" rappresenta il volume di scambio ad apertura dei mercati in quel giorno mentre gli altri attributi danno ulteriori specifiche sul prezzo dello strumento finanziario) utili per costruire la tabella "valori\_giornalieri\_azione", entità che sarà fondamentale per disegnare i grafici.

**Problematiche:** Purtroppo però, il web-service in questione ha delle limitazioni. Gli utenti con un API key posso fare un massimo di 5 richieste al minuto e 500 al giorno. Quindi, di conseguenza, se vogliamo aggiornare più di cinque azioni contemporaneamente riceveremo un messaggio di errore. Questa incombenza è risolvibile acquistando libertà aggiuntive per l'ottenimento di dati.

### api.markestak

La seconda richiesta si riferisce a markestark e il suo URL si presenta nella seguente forma:

```
http://api.marketstack.com/v1/tickers?symbols=$simbolo&access_key=$token
```

Dove “\$simbolo” rappresenta la variabile del simbolo dell’azione che inserisce l’utente (es: GOOG) e “\$token” è la API key ottenuta iscrivendosi al web-service (es: 7636d77b2e219a1319e7f9b167abf4b7). Considerando il formato json e l’azione Alphabet (GOOG) il risultato è del tipo:

```
1 {
2     "name": "Alphabet Inc - Class C",
3     "symbol": "GOOG",
4     "has_intraday": false,
5     "has_eod": true,
6     "country": null,
7     "stock_exchange": {
8         "name": "NASDAQ Stock Exchange",
9         "acronym": "NASDAQ",
10        "mic": "XNAS",
11        "country": "USA",
12        "country_code": "US",
13        "city": "New York",
14        "website": "WWW.NASDAQ.COM"
15    }
16 }
```

Il dato che estrapoliamo dal seguente script in realtà è solo il nome. La restante parte rappresenta dei dati aggiuntivi che sebbene per il momento non vengono utilizzati in un futuro sviluppo potrebbero assumere un’importanza strategica.

Per convertire i dati dal formato json in un formato comprensibile al php si utilizza la libreria cURL che permette non solo di ottenere i dati dai server remoti ma anche inserirli direttamente nel DB. Tra i principali è importante ricordare `$curl = curl_init()` per inizializzare la sessione di cURL, `CURLOPT_URL = linkapi`, `$response = curl_exec($curl)` per eseguire la richiesta, `$data = json_decode($response, true)` per convertire i dati ottenuti e `curl_close($curl)` per chiudere la sessione.

## 2.7 Specifiche di sicurezza

Nello sviluppo del progetto e durante le varie fasi di realizzazione, sono state anche indirizzate le necessità di creare un applicativo sicuro da tutti i punti di vista. Oltre alle sessioni e ai cookies, elementi che permettono di autenticare l’utente, si considera l’utilizzo di un token come strumento per far fronte agli attacchi di CSFR (Cross Site Request Forgery) nel ambito del phishing.

### 2.7.1 Cookies

Il primo livello di sicurezza implementato è quello inerente ai cookies. Un cookie è un file che il server invia al client scrivendoci dentro delle informazioni. Successivamente queste informazioni possono essere lette dal server quando il client richiede una determinata pagina. Nel seguente progetto vengono utilizzati per identificare gli utenti. Il cookie viene creato e settato nel momento del login e poi verificato in tutte le altre aree riservate. Se il cookie risultasse scaduto o non valido automaticamente l’applicativo provvederà a reindirizzare l’utente alla pagina di login. L’interfaccia sarà esattamente quella mostrata in Figura 2.2.

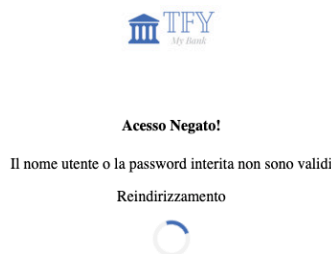


Figura 2.2: Re-indirizzamento



Il cookie ha come caratteristiche preimpostate una durata pari a 10 minuti, riconosciuta anche come (*setcookie("login", "OK", time() + 600)*). Scaduto questo lasso di tempo sarà necessario ed obbligatorio rifare il login.

### 2.7.2 Sessioni

Sebbene PHP consenta di memorizzare lo stato utente tra una pagina web e l'altra attraverso i cookie, questo meccanismo presenta problemi di sicurezza. Infatti, poiché i cookie sono memorizzati sul computer dell'utente, è possibile che un utente malintenzionato modifichi facilmente questi dati al fine di compromettere l'intera applicazione. Un modo alternativo per rendere accessibili i dati in maniera permanente attraverso le pagine di un sito web, consiste nell'utilizzare una sessione PHP. Le sessioni sono un insieme di informazioni memorizzate in un file che si trova sul server dove gira la nostra applicazione. Quando viene avviata una sessione, PHP crea un identificatore univoco per quella particolare sessione, denominato PHPSESSID e rappresentato da una stringa casuale generata tramite una funzione di hash. Allo stesso modo, viene creato e memorizzato sul server un file temporaneo all'interno di una directory designata al salvataggio delle sessioni. Quando uno script PHP desidera recuperare il valore da una variabile di sessione, PHP preleva il valore dell'identificatore di sessione PHPSESSID e lo confronta con quello memorizzato sul server per vedere se sono uguali. In caso affermativo la sessione continua, altrimenti viene interrotta. Come per i cookie, una sessione può anche terminare quando l'utente chiude il browser o dopo aver fatto il logout dalla propria area riservata, oppure si interrompe automaticamente dopo un certo periodo di inattività.

Per avviarla, è sufficiente utilizzare la funzione *session\_start()* mentre per terminarla basterà *session\_destroy()* e per eliminare tutte le variabili *session\_unset()*.

### 2.7.3 CSRF Token

Questo paragrafo si dedica invece agli attacchi CSRF (Cross Site Request Forgery). Per comprendere meglio di cosa stiamo parlando ecco un esempio ipotetico di attacco CSRF al nostro conto investimenti online:

- Bob, cliente di Trading For You, si collega a MyBank.com per consultare le azioni del suo conto.
- Bob non fa il logout su MyBank.com e chiude la scheda del browser. La sessione sul sito è ancora attiva.
- Bob continua a navigare con il browser e finisce sul sito di phishing Info Investimenti. Qui, inserisce la propria email e fa clic sul pulsante "Invia" per iscriversi alla newsletter.
- In realtà, Info Investimenti è una copertura e il pulsante "Invia" non esegue nessuna registrazione, ma invia una richiesta al sito MyBank.com tramite la sessione autenticata in precedenza nel browser per fare un'operazione di 10.000 euro su un'azione fasulla.

Questo livello di sicurezza viene affrontato con l'utilizzo del CSRF Token. Un token CSRF è un valore univoco, segreto e imprevedibile generato dall'applicazione lato server (con funzione *md5*) e trasmesso al client in modo tale da essere incluso in una successiva richiesta HTTP effettuata dal client. Quando viene effettuata la richiesta successiva, l'applicazione lato server convalida che la richiesta includa il token previsto e rifiuta la richiesta se il token è mancante o non valido.

I token CSRF possono prevenire gli attacchi CSRF rendendo impossibile per un utente malintenzionato costruire una richiesta HTTP completamente valida adatta per essere fornita a un utente vittima. Poiché l'attaccante non può determinare o prevedere il valore del token CSRF di un utente, non può costruire una richiesta con tutti i parametri necessari affinché l'applicazione onori la richiesta.

Entrando nel concreto, l'utilizzo di questo token prevede due diverse fasi:

1. La creazione; che avviene tramite una funzione *md5*: una funzione hash crittografica, unidirezionale e irreversibile:

```
1 $token = md5(uniqid(rand(), TRUE));
2 $_SESSION['csrf_token'] = $token;
3 $_SESSION['csrf_token_time'] = time();
```

## 2. La validazione e eventuale distruzione:

```
1  if(isset($_POST['csrf_token'])){
2      if($_POST['csrf_token'] === $_SESSION['csrf_token']){
3      }else{
4          $errors[] = "Problema con la verifica del CSRF Token";
5      }
6  }else{
7      $errors[] = "Problema con la verifica del CSRF Token";
8  }
9
10 // CSRF Token tempo
11 $max_time = 60*60*24; // tempo in secondi
12 if(isset($_SESSION['csrf_token_time'])){
13     // uguale al massimo tempo
14     $token_time = $_SESSION['csrf_token_time'];
15     if(($token_time + $max_time) >= time()){
16     }else{
17         // Errore interfaccia e distruzione del CSRF Token
18         $errors[] = "CSRF Token Scaduto";
19         unset($_SESSION['csrf_token']);
20         unset($_SESSION['csrf_token_time']);
21     }
22 }else{
23     // distruzione del CSRF Token
24     unset($_SESSION['csrf_token']);
25     unset($_SESSION['csrf_token_time']);
26 }
```

## 2.8 Form php per le operazioni

Prima di iniziare ad operare con la base di dati è importante costruire un form per permettere all'utente di interagire con l'applicativo. A tal proposito i form creati sono diversi e si riferiscono a: effettuare il login, inserire un'azione, aprire una posizione e interrogare lo storico con i filtri. Si analizza qui di seguito quello dello storico nonché quello richiesto dal problema. La tecnica utilizzata è quella delPostBack. Essa permette di rimandare le variabili del form direttamente alla pagina che le ha generate tramite il submit(ossia premendo il pulsante) e il metodo POST. Il codice corrispettivo è il seguente:

```
1  <?php
2      $data1 = $data2 = $portafoglio = "";
3      //con if si controlla se il metodo sia POST
4      if ($_SERVER["REQUEST_METHOD"] == "POST") {
5          $data1 = $_POST["data1"];
6      }
7      if ($_SERVER["REQUEST_METHOD"] == "POST") {
8          $data2 = $_POST["data2"];
9      }
10     if ($_SERVER["REQUEST_METHOD"] == "POST") {
11         $portafoglio = $_POST["portafoglio"];
12     }
13 }?>
14
15 <div id="page-wrapper" style="min-height: 345px;">
16 <div class="row">
17     <div class="col-lg-12">
18         <h1 class="page-header">Storico </h1>
19     </div>
20     <div class="row">
21         <div class="col-lg-12">
22             <div class="panel panel-default">
23                 <div class="panel-heading">Filtro</div>
24                 <div class="panel-body">
25                     <div class="row">
26                         <div class="col-lg-12">
27                             // si apre il form
28                             <form role="form" method="post">
29                                 <div class="col-lg-6">
30                                     <div class="form-group">
```

```

31         <label>Gestione</label>
32         //casella di riepilogo
33         <select name="portafoglio" class="form-control">
34         //option vuota per resettare i filtri
35         <option></option>
36         //la connessione viene effettuata in un file a parte e richiamato
           con include ()
37         <?php
38             $sql = "SELECT * FROM portafoglio";
39             $result = $db->prepare($sql);
40             $result->execute() or die(print_r($result->errorInfo(), true));
41             $ps = $result->fetchAll(PDO::FETCH_ASSOC);
42             foreach ($ps as $p) {
43                 ?>
44                 //option con la query precedente (stampa ogni nome della gestione e
                   restituisce il suo id in base alla scelta dell'utente)
45                 <option value="<?php echo $p['id'] ?>"> <?php echo $p['nome']
                   ?></option>
46             <?php } ?>
47         </select>
48     </div>
49     //inserimento date da visualizzare
50     <label>Da : </label> &#09 <input type="date" name="data1" value="<?php echo
                   $data1;?>">&#09<label> A : </label> &#09 <input type="date"
                   name="data2" value="<?php echo $data2;?>">
51     <br><br>
52     //bottone submit
53     <input type="submit" class="btn btn-primary" value="Visualizza" />
54 </div>
55 </div>
56 </div>
57 </form>
58 </div>
59 </div>

```

Come si può notare l'interfaccia non viene trascurata. Essa grazie alle classi inserite nel file CSS si presenta come in Figura 2.3.

Figura 2.3: Form - Storico

## 2.9 Specifiche delle operazioni

Le specifiche dell'operazione sono l'aspetto più importante dell'applicazione, sono le funzioni principali richieste dal problema. Innanzitutto è importante definire il concetto di query. Esse permettono di interrogare i dati contenuti nelle tabelle e crearne un output targettato alle proprie necessità. In particolare, vengono utilizzate per visualizzare dati provenienti da più tabelle, scegliere o ordinare i record, modificarli ed eseguire calcoli basati sul linguaggio sql.

A questo proposito, le prossime sezioni elaboreranno le azioni richieste dal problema.

### 2.9.1 Visualizza operazioni del cliente x

- gestire l'accesso autenticato; qui si vogliono visualizzare esclusivamente le operazioni dell'utente loggato. Lo script php una volta che il cliente è entrato nella piattaforma salva nella variabile “\$\_SESSION['idconto']” l'id del conto corrispettivo. Questo si ottiene grazie alla seguente query:

```
1 $sql = "SELECT conto.id, conto.idcliente, cliente.id, cliente.email
2       FROM conto, cliente
3       WHERE cliente.email= $email AND cliente.id=conto.idcliente ";
```

Dove con SELECT si selezionano gli attributi che si vogliono visualizzare, con FROM la tabella in cui si trovano e con “\$email” l'username inserito nel form dal cliente.

Logicamente la query sopra descritta può essere rappresentata con il diagramma di Venn sottostante (Figura 2.4).

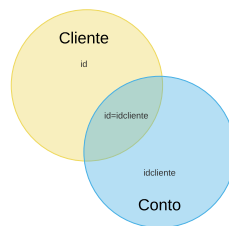


Figura 2.4: Diagramma di Venn - Autenticazione

### 2.9.2 Visualizza operazioni dalla più recente

- ordinare la visualizzazione delle operazioni cronologicamente (dalla più recente):

```
1 $sql = "SELECT s.id, s.nome, s.simbolo, s.mercato, ws.idportafoglio
2       FROM azioni AS s JOIN storico AS ws ON s.id=ws.idazione
3       WHERE ( SELECT portafoglio.id, storico.idportafoglio
4             FROM portafoglio, storico AS ws
5             WHERE $_SESSION['idconto']=portafoglio.idconto AND
6                   ws.idportafoglio=portafoglio.id)
7       ORDER BY ws.creato DESC";
```

Dove con SELECT si selezionano gli attributi che si vogliono visualizzare (id azione, nome società, simbolo alfanumerico dell'azione, la borsa valori) e con FROM la tabella in cui si trovano (azioni e storico). Come si può notare si utilizza la clausola JOIN per combinare le tuple delle tabelle azioni e storico nel momento in cui id e id azione coincidano. In modo similare si utilizza una subquery per filtrare i dati in base all'utente che sta visualizzando.

Il diagramma di Venn si presenta come in Figura 2.5.

La query include anche la clausola ORDER BY [...] DESC che ci permetterà di completare la prima task ordinando i dati cronologicamente dalla più recente (in base all'attributo “creato” che corrisponde alla data di apertura della posizione). La clausola AS invece permette di rinominare le tabelle così da semplificare la comprensione.



Figura 2.5: Diagramma di Venn - Storico

### 2.9.3 Filtro in base all'anno

- visualizzare le informazioni dettagliate delle operazioni di acquisto e di vendita con il relativo saldo dato l'anno:

```

1 $sql = "SELECT s.id, s.nome, s.simbolo, s.mercato
2       FROM azioni AS s JOIN storico AS ws ON s.id=ws.idazione
3       WHERE (SELECT portafoglio.id, storico.idportafoglio
4             FROM portafoglio, storico AS ws
5             WHERE $_SESSION['idconto']=portafoglio.idconto AND
6               ws.idportafoglio=portafoglio.id)
7       AND ws.creato BETWEEN '". $data1. "' AND '". $data2. "'
      ORDER BY ws.creato DESC";

```

Nel costrutto WHERE, si aggiunge la clausola BETWEEN ... AND che permette di recuperare i valori di “creato” compresi tra i due valori inseriti dall'utente (\$data1 e \$data2).

### 2.9.4 Filtro in base alla gestione

- come al punto precedente, ma l'utente può filtrare in base alla procedura di gestione (ovvero “standard”, “auto”, “assisted”):

```

1 $sql = "SELECT s.id, s.nome, s.simbolo, s.mercato
2       FROM azioni AS s JOIN storico AS ws ON s.id=ws.idazione
3       WHERE (SELECT portafoglio.id, storico.idportafoglio
4             FROM portafoglio, storico AS ws
5             WHERE $_SESSION['idconto']=portafoglio.idconto AND
6               ws.idportafoglio=portafoglio.id)
7       AND ws.idportafoglio LIKE '". $portafoglio. "'
8       AND ws.creato BETWEEN '". $data1. "' AND '". $data2. "'
      ORDER BY ws.creato DESC";

```

Dove si aggiunge ancora la clausola LIKE che permette di filtrare le posizioni in base alla procedura di gestione inserita nell'apposito form grazie alla casella di riepilogo (es: \$portafoglio = Auto) .

### 2.9.5 Altro

In aggiunta, sebbene non richieste, si rendono necessarie anche le query relative a:

- visualizzazione delle azioni presenti (con i dati associati)

```

1 //per le azioni
2 $sql = "SELECT *
3       FROM azioni";
4 //per i dati delle stesse
5 $sql = "SELECT *
6       FROM valori_cache_azione
7       WHERE idazione=?";

```

In questo caso sono necessarie due query separate. Questo è dovuto principalmente alle regole di normalizzazione applicate precedentemente.

- visualizzazione delle tipologie di gestione (Auto, Standard, Assisted)

```
1 $sql = "SELECT *
2     FROM portafoglio";
```

Dove si selezionano tutti i campi con il simbolo “\*”.

- visualizzazione delle posizioni aperte nelle gestioni

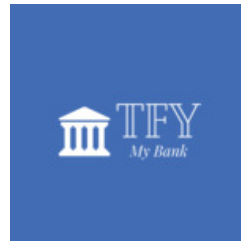
```
1 $sql = "SELECT s.id, s.nome, s.simbolo, s.mercato
2     FROM azioni AS s JOIN portafoglio_azioni AS ws ON s.id=ws.idazione
3     WHERE ws.idportafoglio=?
4     AND   ( SELECT portafoglio.id, storico.idportafoglio
5             FROM portafoglio,storico AS ws
6             WHERE $_SESSION['idconto']=portafoglio.idconto AND
7                   ws.idportafoglio=portafoglio.id)
8     ORDER BY ws.creato DESC";
```

Dove viene eseguita una JOIN come nel caso precedente ma considerando le tabelle “azioni” e “portafoglio\_azioni”.

## 2.10 Sviluppo e descrizione

Il sito web è stato progettato con l’obiettivo di essere semplice, intuitivo e sicuro da tutti i punti di vista. L’efficienza e la linearità sono le caratteristiche principali. A tal punto si è scelto un colore dominante, l’azzurro non solo presente nel logo ma anche nei dettagli delle varie sezioni del sito. Questo colore non è frutto di una scelta casuale ma è stato scelto poiché nel marketing è utilizzato in situazioni di comunicazione nelle quali si vuole trasmettere serietà, fiducia e continuità. Ne sono un esempio le istituzioni europee, EBA (European Banking Authority), CE (Commissione Europea) e ECB (European Central Bank)

Tra i svariati tentativi, sono stati scelti due loghi con stesso design ma a colori invertiti, come .



Il menu di navigazione nella pagina principale è stato studiato anch’esso per poter essere gestito in maniera semplice, immediata e nel contempo fruibile. Tutti i contenuti sono stati suddivisi in sezioni o aree per una corretta ed intuitiva visualizzazione. A ciascuna sezione sono state associate delle pagine comprensive con a volte sotto menu sviluppati all’interno delle pagine dell’applicazione.

Abbiamo quattro sezioni o aree principali e quattro sottosezioni:

1. Area cliente;
2. Ricerca titoli;
  - (a) Aggiungi
  - (b) Visualizza
3. Nuovi dati;
4. Portafoglio;
  - (a) Sezioni
  - (b) Storico

La sezione “Area cliente” è quell’area di front-office dedicata alla visualizzazione delle informazioni dell’utente e del consulente assegnato. Essa, come anche le altre pagine è accessibile solo dopo aver fatto l’accesso tramite un login form.

Le due interfacce si presentano così:

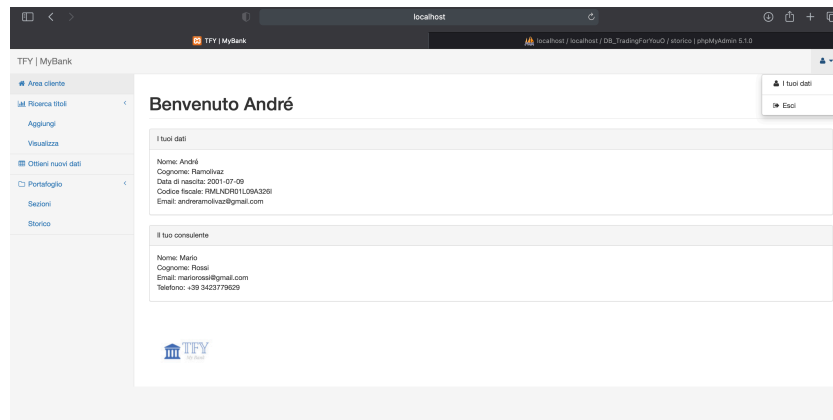


Figura 2.6: Area cliente

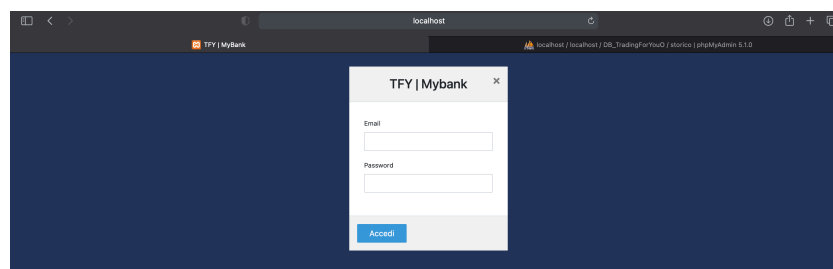


Figura 2.7: Login

La pagina di login è dotata di appositi controlli per cui se l’utente non inserisce nulla si riceve un messaggio di errore.

La sezione “Ricerca Titoli”, invece, permette attraverso i sotto menu ‘Aggiungi’ e ‘Visualizza’ di gestire le informazioni inerenti alle azioni che si vogliono trattare. Esse rappresentano una sorta di watchlist con la quale la gestione *Auto* o *Standard* farà le ipotetiche analisi e consiglierà al cliente le operazioni da fare. I dati da gestire sono principalmente due: la borsa valori su cui si opera, che viene inserita automaticamente da una casella di riepilogo e il codice alfanumerico dell’azione che dev’essere inserito dall’utente.

In tal merito si visualizzano le seguenti pagine:

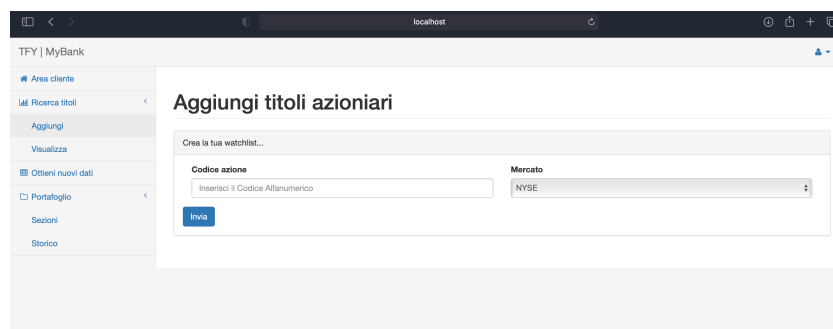


Figura 2.8: Aggiungi

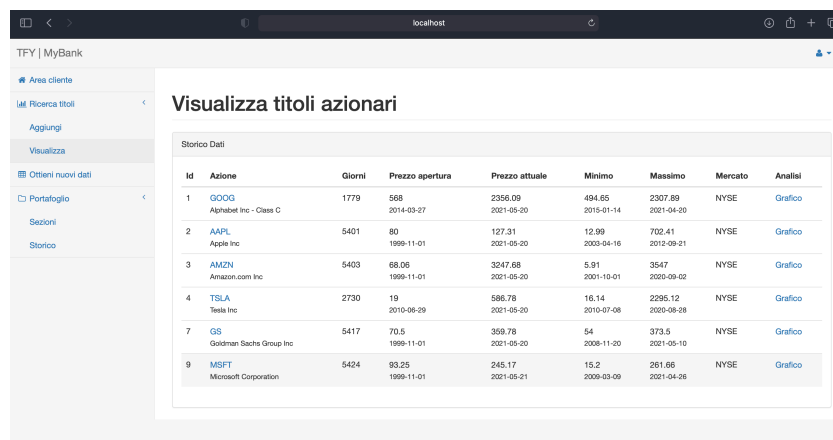


Figura 2.9: Visualizza

Da “Visualizza” si fanno dei collegamenti ipertestuali e si accede ad altre due pagine, quella inerente ai dati giornalieri dell’azione e una seconda per il rispettivo grafico. Utilizzando l’esempio dell’azione Alphabet avremo:

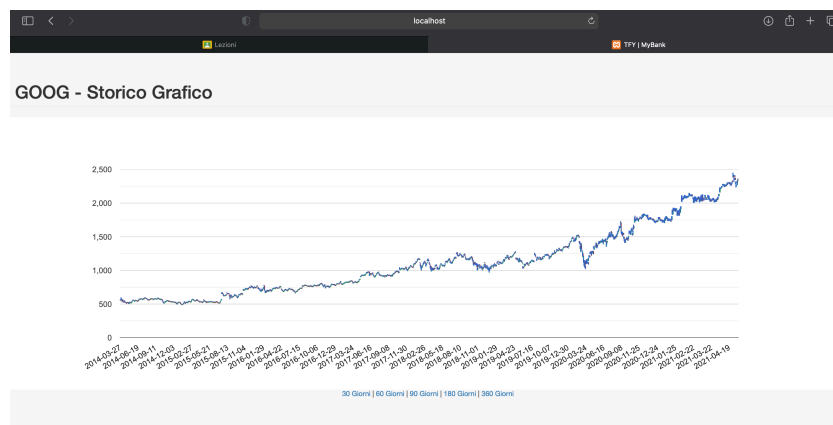


Figura 2.10: Grafico



| Storico dati |        |            |           |           |           |           |         |
|--------------|--------|------------|-----------|-----------|-----------|-----------|---------|
| Id           | Azione | Giorno     | Apertura  | Massimo   | Minimo    | Chiusura  | Volume  |
| 1            | GOOG   | 2021-05-21 | 2365.9900 | 2368.9999 | 2342.3700 | 2345.1000 | 1141632 |
| 1            | GOOG   | 2021-05-20 | 2328.0400 | 2360.3400 | 2321.0900 | 2356.0900 | 1191593 |
| 1            | GOOG   | 2021-05-19 | 2264.4000 | 2316.7600 | 2263.5200 | 2308.7100 | 967544  |
| 1            | GOOG   | 2021-05-18 | 2336.9056 | 2343.1500 | 2303.1600 | 2303.4300 | 865119  |
| 1            | GOOG   | 2021-05-17 | 2309.3200 | 2323.3400 | 2295.0000 | 2321.4100 | 982416  |
| 1            | GOOG   | 2021-05-14 | 2291.8300 | 2321.1400 | 2283.3200 | 2316.1600 | 1331248 |
| 1            | GOOG   | 2021-05-13 | 2261.0900 | 2276.6012 | 2242.7200 | 2261.8700 | 1330861 |
| 1            | GOOG   | 2021-05-12 | 2261.7100 | 2285.3700 | 2230.0500 | 2239.0800 | 1746664 |
| 1            | GOOG   | 2021-05-11 | 2291.8600 | 2322.0000 | 2283.0000 | 2308.7600 | 1605548 |
| 1            | GOOG   | 2021-05-10 | 2374.8900 | 2378.0000 | 2334.7300 | 2341.6600 | 1302071 |
| 1            | GOOG   | 2021-05-07 | 2400.0000 | 2416.4100 | 2390.0000 | 2398.6900 | 1163892 |
| 1            | GOOG   | 2021-05-06 | 2350.6400 | 2382.7100 | 2342.5382 | 2361.3500 | 1020959 |
| 1            | GOOG   | 2021-05-05 | 2368.4200 | 2382.2000 | 2351.4100 | 2366.7400 | 1096275 |
| 1            | GOOG   | 2021-05-04 | 2369.7400 | 2379.2600 | 2311.7000 | 2364.2500 | 1758215 |
| 1            | GOOG   | 2021-05-03 | 2402.7200 | 2419.7000 | 2384.5000 | 2395.1700 | 1689389 |
| 1            | GOOG   | 2021-04-30 | 2404.4900 | 2427.1400 | 2402.1600 | 2410.1200 | 1957130 |
| 1            | GOOG   | 2021-04-29 | 2410.3300 | 2436.5200 | 2402.2800 | 2429.8900 | 1977677 |
| 1            | GOOG   | 2021-04-28 | 2407.1450 | 2452.3780 | 2374.8500 | 2378.9100 | 2586439 |
| 1            | GOOG   | 2021-04-27 | 2336.0000 | 2337.4500 | 2304.2700 | 2307.1200 | 1588583 |

Figura 2.11: Dati giornalieri

La sezione “Nuovi Dati”, al contrario, è poco grafica. La sua funzione è quella di aggiornare i dati giornalieri delle azioni e mostra un semplice messaggio in caso di successo. Il tempo per l’aggiornamento è abbastanza lungo ma è normale perché il php controlla tutti dati e aggiorna solo quelli mancanti per non creare ridondanze.

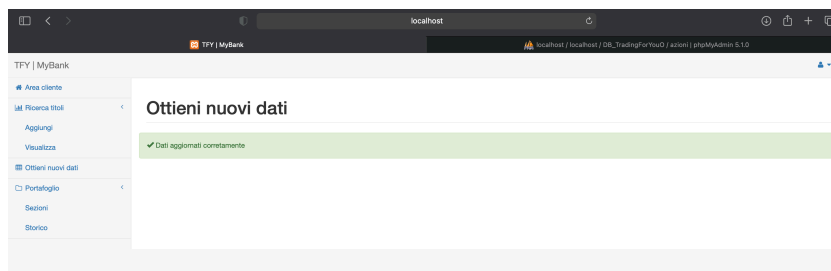


Figura 2.12: Nuovi Dati

La sezione “Portafoglio”, invece, permette attraverso i sotto menu ‘Sezioni’ e ‘Storico’ di gestire l’apertura e chiusura delle azioni. Come il testo richiede, ci sono tre gestioni: Auto, Standard e Assisted. Si presuppone che nelle sezioni Auto e Assisted ci sia un software dedicato alla gestione automatica (compra e vendi), l’utente deve interagire solo con la sezione Standard. Essa permette di selezionare la tipologia di operazione (compra/vendi tramite check list), la quantità di azioni e il nome dell’azione su cui si vuole operare (che dev’essere inserita prima nel Data Base tramite casella di riepilogo). Si visualizzano inoltre le posizioni aperte e il saldo complessivo delle stesse. Si visualizzano pertanto le seguenti pagine:

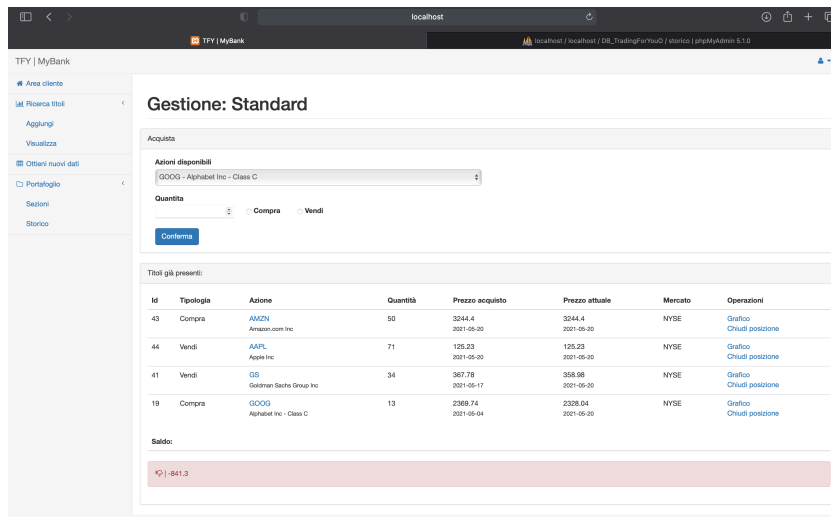


Figura 2.13: Gestione sezione

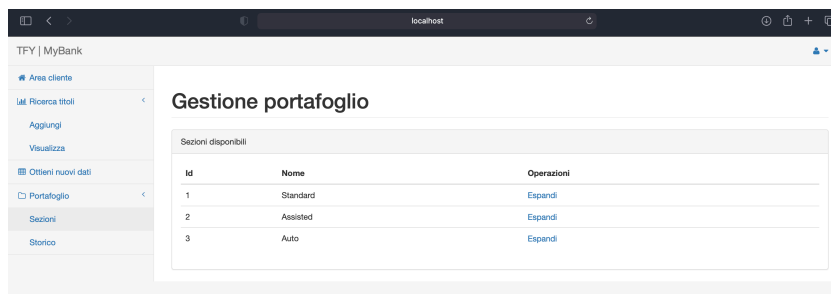


Figura 2.14: Sezioni

Infine, l'ultima area è quella dello storico. Essa si presenta analoga alla gestione delle sezioni tranne che per l'impossibilità di acquistare o vendere. Il saldo complessivo non è statico bensì varia in base ai filtri utilizzati che altro non sono che derivati delle query prima presentate. Si ottiene la pagina seguente:

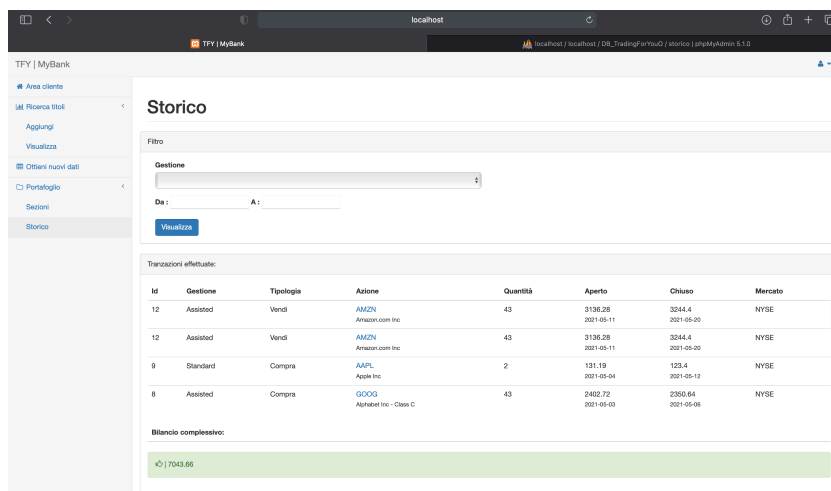


Figura 2.15: Storico

# Capitolo 3

## Sistemi e Reti

Questo capitolo, a seguito di un piccolo approfondimento sulle tecnologie e strumenti utilizzati, si dedica alla struttura di rete (evidenziandone l'utilità e l'applicabilità nell'ambito proposto), alla parte relativa alla autenticità e sicurezza di quest'ultima e ai diversi protocolli, dispositivi e canali che si possono proporre.

### 3.1 Strumenti utilizzati

Per questo capitolo si utilizza uno dei più diffusi simulatori di rete: il Packet Tracer. Il software consente agli utenti di creare topologie di rete, simulare la configurazione di router e switch Cisco utilizzando un'interfaccia a linea di comando simulata.

### 3.2 Raccolta e analisi dei requisiti

Prima di andare nello specifico di Packet Tracer sono stati raggruppati nel glossario seguente alcuni termini che saranno utili per questa fase:

| Termine      | Significato  |
|--------------|--|
| LAN          | Rete costituita da computer collegati tra loro, dalle interconnessioni e dalle periferiche condivise in un ambito fisico delimitato. Tradizionalmente lavorano tra 10 Mbps e 100 Mbps, e sono caratterizzate da bassi ritardi e pochissimi errori. |
| Server       | Elemento informatico e delle telecomunicazioni che elabora e gestisce le informazioni su una rete, restituendole a tutti coloro che ne fanno richiesta (chiamati client).  |
| Router       | Dispositivo elettronico che, in una rete informatica, si occupa di instradare i dati, suddivisi in pacchetti, fra reti diverse.  |
| Switch       | Dispositivo di comunicazione situato al livello data link (livello2) dell'OSI Model che inoltra il traffico su una LAN basandosi sull'indirizzo MAC.   |
| Access Point | Ricetrasmittitore che usa onde radio per connettere reti cablate con dispositivi wireless o reti esclusivamente wireless.  |
| Firewall     | Router che consente di configurare le regole in entrata corrispondenti agli indirizzi IP di origine / destinazione e ai numeri di porta locale / remota. In base alla corrispondenza, la connessione può essere consentita o negata.               |
| Cloud PT     | Dispositivo che si presenta come una nuvola nello schema e rappresenta la connessione a Internet.  |

**Tabella 3.1 continuata dalla pagina precedente**

|                              |   |
|------------------------------|---|
| <b>DHCP</b>                  | (Dynamic Host Configuration Protocol) Un un protocollo caratterizzato da un server DHCP il cui compito è di assegnare ad un dispositivo che si connette alla sua rete il primo indirizzo IP valido disponibile. |
| <b>NAT</b>                   | Una tecnica attuata dal router che, nell'intestazione di un pacchetto IP sostituisce un indirizzo, sorgente o destinazione con un altro indirizzo.  |
| <b>DNS</b>                   | Servizio che gestisce la traduzione di nomi in indirizzi IP, controllando a quale server si conatterà un utente finale nel momento in cui digiterà un nome di dominio nel suo browser Web.                      |
| <b>AAA</b>                   | (Authorization Authentication Accounting) Servizio per effettuare l'autenticazione e l'autorizzazione degli utenti, nonché l'accounting degli accessi.  |
| <b>DataBase server</b>       | Server essenziale per il funzionamento del Web. Ha il compito di mettere a disposizione le sue risorse sulla rete.  |
| <b>Webserver</b>             | Server in grado di gestire le richieste di trasferimento di pagine web di un client, tipicamente un browser.  |
| <b>Protocollo TCP</b>        | (Transmission Control Protocol) Protocollo che gestisce il flusso delle informazioni fra due nodi e ne garantisce l'arrivo dei dati.  |
| <b>Protocollo IP</b>         | Protocollo a pacchetti di interconnessione di reti, classificato al livello di rete del modello ISO/OSI, nato per interconnettere reti eterogenee.  |
| <b>Protocollo HTTPS</b>      | Protocollo per la comunicazione su Internet che protegge l'integrità e la riservatezza dei dati scambiati tra i computer e i siti.  |
| <b>IEEE 802.3 o Ethernet</b> | Insieme di specifiche tecniche che a loro volta definiscono una tecnologia standard per reti locali (LAN).  |
| <b>IEEE 802.11</b>           | Insieme di standard di trasmissione per reti WLAN.  |

Tabella 3.1: Glossario

### 3.3 Premessa

La rete funziona collegando computer e periferiche tramite switch, router e punti di accesso. Questi dispositivi sono le basi essenziali di networking, che consentono ai vari dispositivi collegati alla rete di comunicare tra loro, così come con altre reti.

### 3.4 Schema di rete (Sede Trading For You)

In base alle specifiche richieste l'infrastruttura di rete è riassumibile nello schema logico riportato in Figura 3.1 <sup>1</sup>:

<sup>1</sup>Le simulazioni in packet tracer sono state realizzate solo a scopo illustrativo.

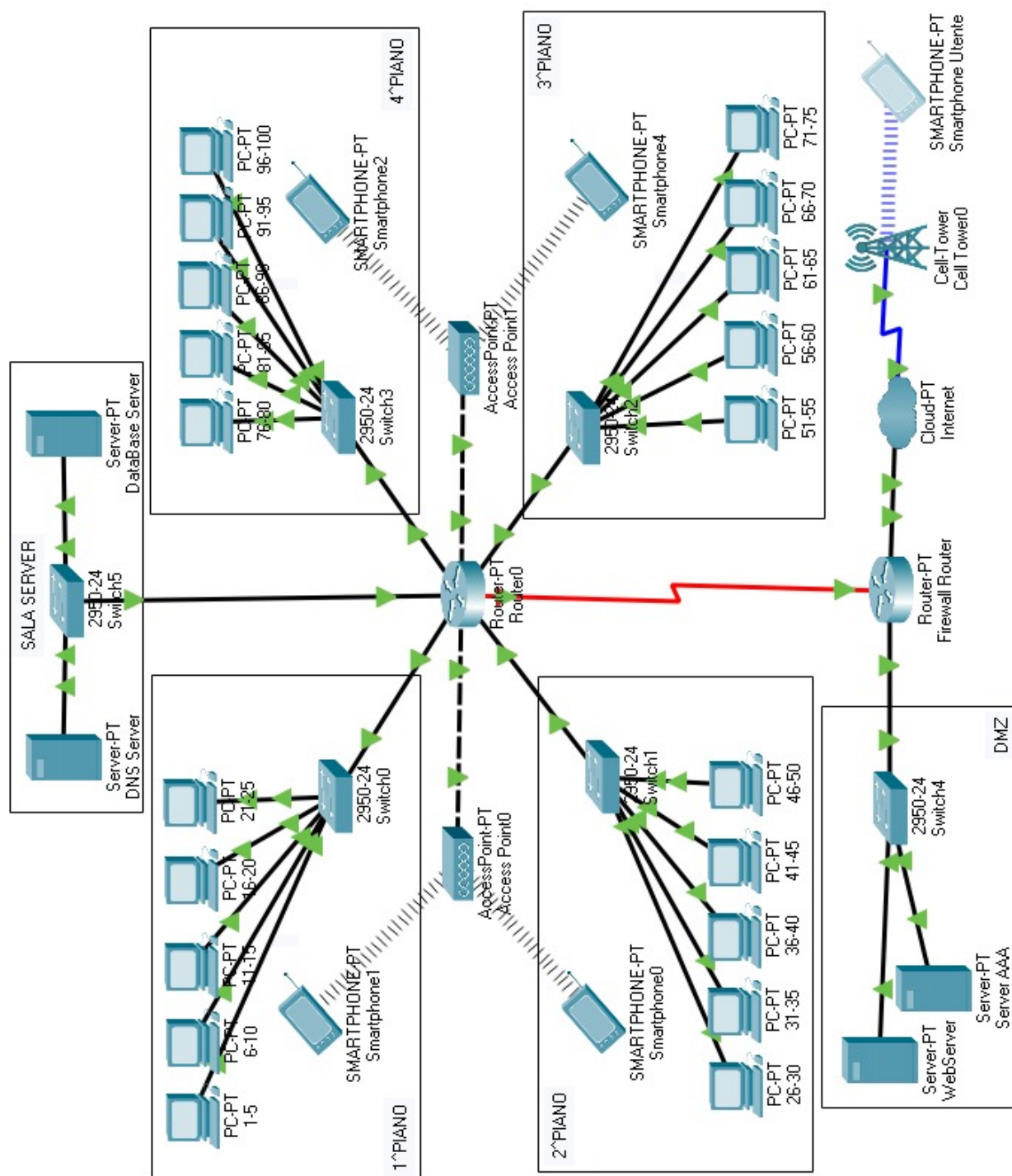


Figura 3.1: Progettazione rete - Packet Tracer

### 3.4.1 Descrizione generale

Lo schema rappresenta sia il sistema della sede centrale che il sistema per i clienti che vogliono operare sulla piattaforma.

Nella sede centrale opera il personale di Easy Train che gestisce il portale. Per svolgere il lavoro in ogni piano sono presenti 25 postazioni fisse che sono collegate alla parta cablata della LAN interna. E' comunque possibile utilizzare la connessione WiFi in quanto si prevede l'installazione di Acces Point ogni due piani (che avranno lo stesso SSID e la stessa identica chiave WEP o WPA in modo da permettere ad eventuali dipendenti di non dover cambiare nulla, se si spostano da un piano all'altro).

La rete, a stella estesa, presenta un centro stella principale rappresentato dal "Router0". Il suo compito è duplice: separare fisicamente la rete dell'azienda in sei aree distinte (la DMZ, i vari piani, la sala server) e collegare la LAN a Intenet. Per fare questo si utilizza un "Firewall Router" con due porte LAN-Ethernet, le quali si riferiscono alla DMZ e alla rete dei vari piani. Inoltre ha una interfaccia WAN-PPP(Point-to-Point Protocol) per il collegamento con il provider che deve essere dotata di tecnologia FTTC (Fiber To The Cab), cioè con fibra ottica.

La rete si compone poi di 4 switch, uno per ogni rete segmentata. Essi rappresentano i centri stella secondari per collegare la DMZ, la sala server e le postazioni fisse. Ogni due piani è presente un Access Point dual band (2.4GHz e 5GHz) conforme allo standard 802.11a. Le prestazioni di questi access point sono grado di gestire l'accesso alla rete di 100 dipendenti contemporaneamente e grazie a dei ripetitori potenti di coprire due interi piani.

I server AAA e Webserver sono raggruppati in un area a parte a garanzia di una maggior sicurezza per i dati sensibili e servizi dell'azienda. Il "Firewall Router" con funzionalità NAT effettuerà la traslazione degli indirizzi pubblico/privato dei pacchetti in entrata e in uscita dalla rete. DMZ, firewall NAT e server AAA rappresentano la protezione della rete dagli attacchi esterni. Infatti nel caso in cui un servizio in LAN sia compromesso in seguito a una vulnerabilità, l'aggressore potrebbe raggiungere anche gli altri host della rete, dato che in LAN non esiste isolamento tra il server e gli altri nodi. Tuttavia, nel caso in cui lo stesso problema si verificasse in DMZ, l'aggressore avrebbe grosse difficoltà a raggiungere la LAN, poiché il traffico tra i server front-end e back-end è fortemente limitato dal firewall. Non disporre di un firewall significa essere esposti a numerosi attacchi e tentativi di intrusione. Nel caso di un semplice sistema casalingo, il danno può essere minimo, ma nel caso di un'azienda come "TradingForYou", una perdita di dati può risultare un danno importante in termini di costi e affidabilità. Il firewall diventa quindi uno degli strumenti più efficaci per la gestione della sicurezza delle reti, grazie a meccanismi di controllo degli accessi, insieme con la possibilità di gestire le regole per la sicurezza.

Per quanto riguarda i programmi server (come il DHCP, e-mail, DNS e DB) si installano su server dedicati inseriti in una sala apposita (che si ipotizza sia al piano terra) dotata di appositi impianti di climatizzazione e di controllo ambientale che includono sistemi di raffreddamento, impianti di antincendio e sistemi di sicurezza avanzati. Oltre a ciò si può considerare di inserire un server di backup o gruppi di continuità (UPS centers) per garantire la continuità dei servizi.

In aggiunta si propone l'algoritmo RSA a chiave asimmetrica, e in particolare l'utilizzo di un token USB come strumento implementativo per la sicurezza. L'applicazione di firma provvederà a richiedere l'inserimento del PIN di protezione che potrà poi essere verificato da servizi secondari come quello di InfoCert o di PosteCert.

#### SINTETIZZANDO:

|             |                   |  |
|-------------|-------------------|--|
| Dispositivi | Acces Point       |  |
|             | Switch            |  |
|             | Router            |  |
|             | Server vari       |  |
|             | Postazioni fisse  |  |
| Canali      | Wirless aziendale |  |

**Tabella 3.2 continuata dalla pagina precedente**

|            |                     |                                      |
|------------|---------------------|--------------------------------------|
| Servizi    | Internet            |                                      |
|            | Cablaggio           |                                      |
|            | DHCP                |                                      |
|            | NAT                 |                                      |
|            | DNS                 |                                      |
|            | AAA                 |                                      |
|            | DB                  |                                      |
|            | Web                 |                                      |
| Protocolli | Livello Data/Link   | IEEE 802.11<br>IEEE 802.3 o Ethernet |
|            | Livello Transport   | TCP                                  |
|            | Livello Network     | IP                                   |
|            | Livello Application | HTTPS                                |

Tabella 3.2

### 3.5 Accesso al sistema per le operazioni

Al fine di consentire l'accesso al sistema mediante l'utilizzo di uno smartphone per l'utente è necessario utilizzare la rete cellulare. La rete cellulare è realizzata per mezzo di onde radio e ricetrasmittitori terrestri (cioè antenne ubicate sulla superficie terrestre), che danno vita alle Base Station (BS), le quali coprono ciascuna una porzione di territorio detta cella di copertura. L'antenna riceve i segnali dagli MU (Mobile User), cioè dai dispositivi mobili degli utenti, e li trasmette agli MSC (Mobile Switching Center), ovvero le centraline che, oltre a connettere tra loro le celle, permettono il collegamento alla rete telefonica fissa e alla rete Internet agendo da gateway.

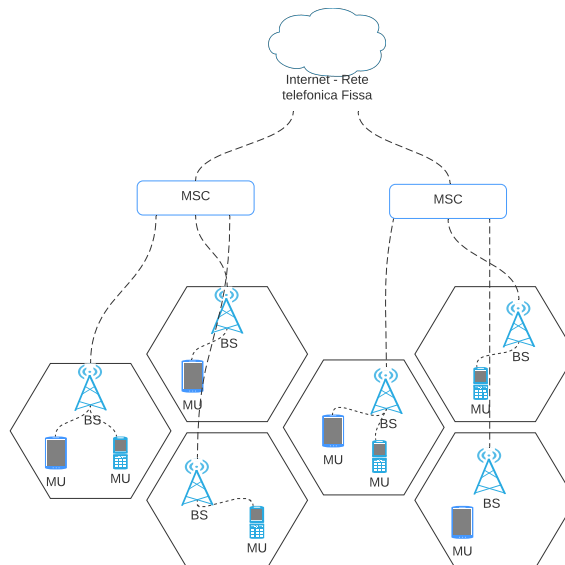


Figura 3.2: Rete cellulare

Ogni rete cellulare è composta da un numero variabile di celle che permettono la copertura radioelettrica e il collegamento tra i terminali mobili e la rete telefonica fissa. Il numero di celle e la loro grandezza dipende dalla quantità di traffico(e quindi dal numero di terminali mobili) stimata in una data zona. Nelle città il numero delle celle è molto alto e la loro grandezza è molto ridotta.

Sul territorio sono presenti più reti cellulari, gestite da operatori diversi come Tim, Vodafone e Wind. Con i sistemi cellulari si ricorre alla tecnica del riutilizzo delle frequenze, cioè, una stessa frequenza è utilizzata più volte ma in celle diverse, sufficientemente lontane tra loro in modo da evitare interferenze:

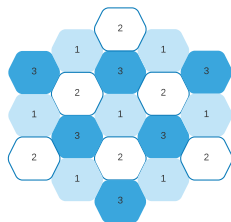


Figura 3.3: Celle adiacenti con frequenze diversi

Il problema principale di questa tecnologia sta nell'affrontare il problema di gestione della mobilità, noto come Handoff(chiamato anche handover), strettamente legato all'interruzione nel collegamento di fronte ad un cambio di Base Station, ovvero al cambio di frequenze nell'ambito dello stesso MSC. L'handoff sullo stesso MSC prevede nove passaggi:

1. La vecchia BS informa l'MSC di un imminente handoff e fornisce la lista di una o più nuove BS.
2. MSC instaura l'instradamento verso la nuova BS.
3. La nuova BS alloca i canali radio per il mobile.
4. La nuova BS segnala all'MSC che è pronta e questo a sua volta lo comunica alla vecchia BS.
5. La vecchia BS attiva il mobile per effettuare handoff.
6. Il mobile e la nuova BS si scambiano i messaggi per completare l'assegnazione del canale.
7. Il mobile comunica alla nuova BS, che a sua volta comunica all'MSC che l'handoff è completato.
8. L'MSC ridireziona le chiamate.
9. La vecchia BS rilascia le risorse.

Un secondo problema che richiede attenzione è legato alla mobilità in caso di cambio di gestore. In questo caso servono specificare i due tipi di rete e i loro database:

- Home Network: la rete del proprio operatore dotata di Home Location Register (HLR) il database contenente le informazioni sull'utente (i.e. numero di telefono, il tipo di contatto, l'attuale posizione).
- Visited Network: la rete presso la quale si trova attualmente il dispositivo dotata del database Visitor Location Register (VLR) il quale ha funzioni analoghe al HLR.

Il processo per questa fase include i seguenti passaggi:

1. La chiamata viene instradata verso la Home Network
2. L'MSC della Home Network accede al suo HLR per ottenere il riferimento del dispositivo mobile presso l'altra rete
3. Tramite la rete telefonica fissa si raggiunge la Visited Network ove si trova in quel momento il dispositivo mobile
4. L'MSC della Visited Network, verifica la presenza dell'utente mobile tramite l'accesso al suo VLR

Per ciò che riguarda la tipologia di rete utilizzata si fa riferimento alla quinta generazione(5G). Quest'ultima permette di costruire reti con bassissima latenza, con elevata quantità di banda e flessibilità nella riconfigurazione. Queste caratteristiche sono possibili grazie a tutta una serie di servizi che possono essere raggruppati in tre grandi categorie:



- servizi mobile ultrabroadband evolution: per l'accesso a Internet a banda ultra larga.
- servizi dell'IoT massivo: per una comunicazione di poche informazioni e non particolarmente sensibili al ritardo.
- servizi critici: per aree come la nostra che richiedono altissima affidabilità, latenza ridotta, elevati requisiti di sicurezza e disponibilità di servizi.

**SINTETIZZANDO:**

|             |                                 |
|-------------|---------------------------------|
| Dispositivi | BS                              |
|             | MSC                             |
|             | MU                              |
| Canali      | Rete cellulare                  |
| Servizi     | Handoff                         |
|             | HLR                             |
|             | VLR                             |
|             | Mobile ultrabroadband evolution |
|             | IoT massivo                     |
| Protocolli  | IP Mobile                       |

Tabella 3.3

# Capitolo 4

## Conclusioni

Per concludere questo l'elaborato, si vuole condividere l'esperienza vissuta e una SWOT analisi sul caso analizzata che si conclude con una proposta concreta su possibili futuri sviluppi della *Trading For You*.

Gli aspetti più interessanti del progetto sono stati la gestione della sicurezza e il cimentarsi in modo autonomo ad un progetto dall'inizio alla fine in ogni suo dettaglio. Questi mi hanno fornito la possibilità di sviluppare specifiche soft skills organizzative e di problem solving ed allo stesso tempo mettere in pratica la teoria acquisita negli ultimi anni di scuola superiore, in particolar modo in merito alla programmazione che trovo più appassionante. Inoltre la combinazione tra l'informatica e il trading online credo sia un ottimo esempio di come la trasformazione tecnologica abbia un ruolo cruciale e facilitante ad oggi che diventerà sempre più essenziale e sfruttato in futuro.

### 4.1 SWOT Analysis

Si vuole concludere questo lavoro con una classica analisi SWOT<sup>1</sup> che illustra con semplicità gli aspetti cruciali dei risultati ottenuti. Come si evince dall'infografica in Figura 4.1 il caso è caratterizzato da notevoli opportunità e punti di forza che sono più rilevanti rispetto alle minacce e punti di debolezza che fanno capo principalmente alla Cyber security e al funzionamento efficiente di fattori esterni. Vediamo ora più nel dettaglio le varie aree.

#### **Strenghts** (punti di forza interni)

- Velocità e precisione: l'applicativo web è flessibile e agile grazie alla linearità e semplicità della struttura creata e realizzata.
- Efficienza: obiettivo finale viene raggiunto e soddisfa tutte le possibili richieste dell'utente permettendone anche sviluppi futuri senza creare limitazioni.
- Sicurezza: diversi regimi di sicurezza sono stati inseriti sia di rete che di autenticazione per proteggere l'utente e prevenire eventuali attacchi esterni.
- Automatizzazione dati: i dati non sono copiati e incollati dall'utente ma vengono resi disponibili con un processo automatizzato (richiesta API). Questo non solo aggiunge nuove informazioni ma le modifica se necessario in caso di correzioni. Cosa ancora più importante, viene mantenuto uno storico a cui si aggiungono i nuovi dati.

#### **Weaknesses** (punti di debolezza interni)

- Richiesta API limitata: la versione gratuita del web server da la possibilità di scaricare solo 5 azioni al minuti e 500 a giornata come riporta il seguente errore "Our standard API call frequency is 5 calls per minute and 500 calls per day". Nella versione premium, a pagamento, questo limite può essere superato.

---

<sup>1</sup>Vedi a pagina 43

- Report scarsi: al momento l'applicazione propone dei report grafici e sintetizzanti plain vanilla. L'agilità con cui è stato costruito il sito web agevola la creazione o l'inserimento di report di analisi più complessi e elaborati che propongono risultati di tecniche predittive.
- Dipendenza dalla rete: come quasi ogni sito vi è una dipendenza naturale alle performance della rete Internet. Per questa ragione, un semplice rallentamento potrebbe causare disagi all'user.

#### **Opportunities** (opportunità esterne)

- Algoritmi predittivi: le nuove tecniche di machine learning permettono e supportano la creazione di algoritmi predittivi che studiano e imparano dal comportamento dell'utente e dall'andamento degli strumenti finanziari. A questo proposito si potrebbero inserire degli algoritmi che in base alla versione al rischio dell'utente propongono lo stesso le azioni più in linea con il suo profilo e le sue aspettative. Da una prospettiva diversa, questi possono anche realizzare dei report automatizzati on-demand più specifici legati per esempio su tematiche molto attuali come il rischio climatico o la moneta digitale.
- News sentiment: si potrebbe inserire un box in cui vengono riportate le analisi di sentimento del mercato su specifiche tematiche. Questo non è altro che un web scraping che sta ad indicare la percentuale di sentimento positivo o negativo (identificate da semplici parole chiave) da articoli del settore (Bloomberg, Moody's, SP) e da giornali (Il sole 24 Ore, The Wall Street Journal).
- Apertura a mercati mondiali: nella versione proposta l'impostazione base è tarata sul mercato NYSE. Il sistema però permette di collegarsi ad ogni mercato mondiale dipendendo dalla versione premium della web API.

#### **Threats** (Minacce di matrice esterna)

- Hackeraggio: nonostante le avanzate precauzioni di sicurezza inserite, l'hackeraggio è in costante evoluzione per i suoi metodi e tecniche. E' pertanto sempre opportuno tenere un profilo conservativo e in allerta, pronto ad adeguarsi e aggiornarsi ai nuovi sviluppi.
- Access Point Rouge: questi altro non sono che accesso alla rete aziendale senza autorizzazione. Per esempio, un dipendente che acquisti e installi un'Acces Point non autorizzato privo di adeguata tecnica di crittografia, prenderebbe la connessione alla rete aziendale non protetta rispetto ad un hacker dotato di dispositivo wireless.
- Sovraccarico della rete: anche se non ancora testato, il numero di utenti che possono svolgere numerose operazioni è limitato. Questi potrebbe causare un rallentamento/crash del sistema.

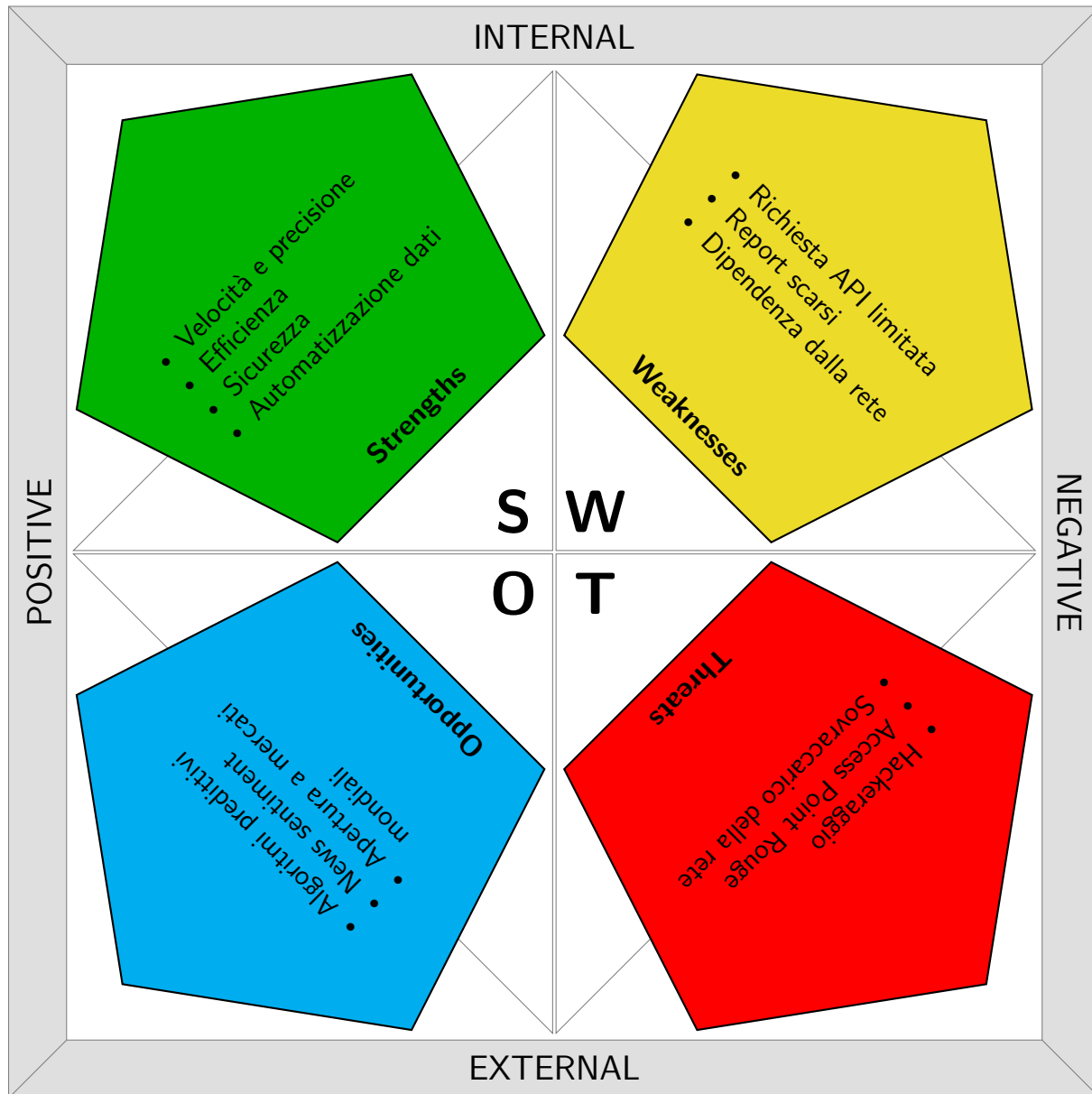


Figura 4.1: Analisi SWOT