

Digiworld
System Design Document



| | |
|--|------------------|
| Progetto: Digiworld | Versione: X.Y |
| Documento: Requirement Analysis Document | Data: GG/MM/AAAA |

Coordinatore del progetto:

| Nome | Matricola |
|--------------------|------------|
| Jacopo Sabino Sole | 0512106112 |

Partecipanti:

| Nome | Matricola |
|--------------------|------------|
| Salvatore Senarcia | 0512106100 |
| Vincenzo Fortino | 0512101240 |
| Sabato Pescatore | 0512105800 |
| Jacopo Sabino Sole | 0512106112 |

Revision History

| Data | Versione | Descrizione | Autore |
|------------|----------|---|--------------------|
| 29/11/2020 | 1.0 | Versione iniziale | [Tutti] |
| 03/12/2020 | 1.1 | Ristrutturazione del diagramma dei servizi | Salvatore Senarcia |
| 10/12/2020 | 1.2 | Aggiornamento divisione in sottosistemi | Sabato Pescatore |
| 15/12/2020 | 1.3 | Ristrutturazione divisione in sottosistemi | Jacopo Sabino Sole |
| 18/12/2020 | 1.4 | Aggiornamento e Ristrutturazione servizi dei sottosistemi | Vincenzo Fortino |
| 28/12/2020 | 1.5 | Aggiornamento schema E-R | [Tutti] |
| 03/01/2021 | 2.0 | Revisione generale con aggiornamento dei servizi dei sottosistemi e della matrice d'accesso | [Tutti] |

Sommario

| | |
|--|----|
| 1.1. Obiettivi del sistema | 3 |
| 1.2. Obiettivi di progettazione | 3 |
| 2. Sistema corrente | 4 |
| 3. Architettura del Sistema Proposto..... | 4 |
| 3.1. Panoramica | 4 |
| 3.2. Decomposizione in Sottosistemi | 5 |
| 3.2.3. Diagramma di deployment | 6 |
| 3.3. Mapping hardware e software..... | 7 |
| 3.4. Gestione dati persistenti | 7 |
| 3.5. Controllo degli accessi e sicurezza | 7 |
| 3.6. Controllo flusso globale del sistema..... | 7 |
| 3.7. Condizioni limite | 8 |
| 3.7.1 Fallimento | 9 |
| 4. Servizi dei Sottosistemi | 10 |

1 Introduzione

1.1. Obiettivi del sistema

La nostra piattaforma e-commerce, DigiWorld, permette di acquistare insieme videogiochi, musica e libri.

Il sistema proposto parte dall'idea di poter fornire supporto ai clienti durante la pandemia globale COVID19. Si vuole quindi realizzare una piattaforma web che consentirà l'acquisto di diversi tipi di categorie tramite un semplice web browser, indipendentemente dal dispositivo utilizzato. DigiWorld avrà la necessità di gestire i dati di ogni profilo del cliente, e la scelta, per motivi di efficienza, ricade su un database relazionale che permetta di poter tener traccia delle informazioni fondamentali necessarie al sistema. Da tale database attingerà un'applicazione web deputata alla gestione delle interazioni con l'utente ed alla manipolazione dei suddetti dati,

La piattaforma garantirà il controllo degli accessi tramite la possibilità di autenticarsi in seguito inserimento di una username e di una password.

1.2. Obiettivi di progettazione

I nostri obiettivi di progettazione per il sistema sono:

Robustezza: Vogliamo proporre un sistema che abbia la capacità di sopravvivere ad input non validi immessi dall'utente. Pertanto, il sistema deve garantire il filtraggio dei dati inconsistenti o errati inseriti dall'utente, invitandolo a reinserirli.

Affidabilità: Il sistema dev'essere in grado di riconoscere situazioni anomale e prevenire modifiche ai dati persistenti al fine di garantirne la consistenza.

Sicurezza: Necessario è fornire uno strumento di autenticazione sicuro, composto dalla richiesta di

username e password prima di ogni accesso ad informazioni riservate così da proteggere i dati da accessi non autorizzati. Le suddette password saranno crittografate.

Costi di sviluppo: Lo sviluppo del prodotto richiederà costi ridotti sia in termini di risorse umane (per cui è fissato un tetto di 75 ore-lavoro), sia in termini economici (per cui si punta a ricorrere a soluzioni off-the-shelf open source).

Tempi di risposta: Il sistema deve elaborare le richieste e produrre output in meno di 2 secondi (al netto di ritardi dovuti alla trasmissione su rete).

Throughput: Il sistema deve permettere l'interazione contemporanea di almeno 100 utenti diversi.

Modificabilità: Le funzionalità del sistema devono essere facilmente modificabili.

Estensibilità: Il sistema deve agevolare l'introduzione di nuove funzionalità.

2. Sistema corrente

Nel nostro caso non è presente un'architettura software già esistente, pertanto abbiamo analizzato le architetture di sistemi simili.

Le piattaforme prese in considerazione sono InstantGaming e Kinguin che offrono la possibilità di poter acquistare prodotti, principalmente giochi, in versione digitale. Tutte e due sono accessibili tramite sito web raggiungibile con qualsiasi web browser.

Ciò che si è evinto dall'analisi è che alla base di tutte le piattaforme è presente un database per la memorizzazione dei dati persistenti interrogato da applicazioni web progettate con diverse tecnologie (rispettivamente PHP, ASP e JSP operanti su server Apache). Deduciamo che ognuna delle suddette piattaforme sia quindi sviluppata secondo un'architettura a 3 strati, ma non siamo in grado di stabilire se siano semplicemente 3 livelli oppure 3 differenti tier.

Per quanto riguarda il controllo degli accessi, quello che si evince è che entrambi i sistemi presentano un'autenticazione con username e password in modo da rendere disponibili le funzionalità che offrono solamente ad utenti autorizzati, mentre non possiamo conoscere i criteri di sicurezza utilizzati per la gestione dei dati sensibili (cifatura delle credenziali o simili).

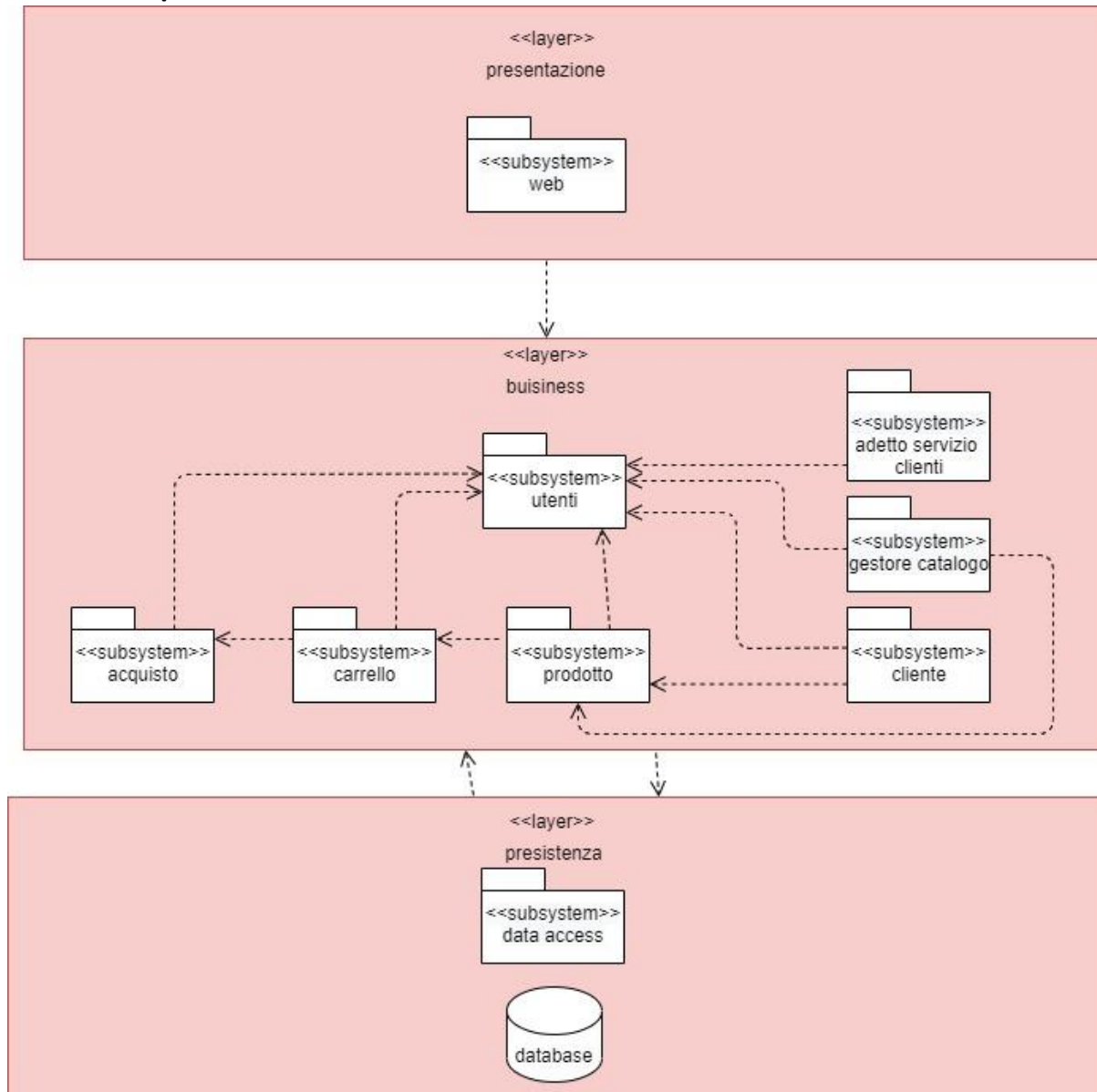
3. Architettura del Sistema Proposto

3.1. Panoramica

La realizzazione del sistema si baserà sull'architettura 3-tier così da rendere la piattaforma web di facile manutenzione e aggiornamento.

- Il primo livello sarà quello di presentazione, che comprende tutte le interfacce grafiche con cui l'utente avrà a che fare. Il tutto sarà molto semplice ed intuitivo per l'utilizzatore e seguirà il pattern MVC. Così facendo potremo distinguere l'interfaccia grafica, la gestione delle richieste e la parte della logica di business.
- Il secondo livello sarà quello logico, che conterrà tutta la business logic del sito.
- Il terzo ed ultimo livello sarà quello dei dati, che si occuperà del collegamento tra la stessa applicazione web ed il database.

3.2. Decomposizione in Sottosistemi



Il sistema è suddiviso in 3 modelli logici: presentazione, business e persistenza. Il livello di presentazione mostra all'utente il sistema; il livello di business si occupa della logica applicativa, infine il livello di persistenza gestisce i dati.

Il livello di presentazione è composto dal sottosistema WEB che si occupa del controllo flusso di esecuzione e mostra l'interfaccia utente.

Il livello di business è composto dai seguenti sottosistemi:

- Utenti: definisce gli utenti generici e permette l'autenticazione
- Cliente: consente l'iscrizione al sito dell'utente, gestisce i clienti, permette di avere una pagina personale
- Addetto servizio clienti: gestisce i problemi dei clienti e i commenti dei prodotti.
- Gestore catalogo: gestisce i prodotti presenti sul catalogo del sito tramite la possibilità di aggiungere eliminare o modificare un prodotto
- Prodotto: modella tutti i prodotti presenti nel catalogo
- Carello: oggetto che raggruppa più prodotti per consentire un acquisto unico
- Acquisto: i prodotti che vengono effettivamente acquistati dall'utente

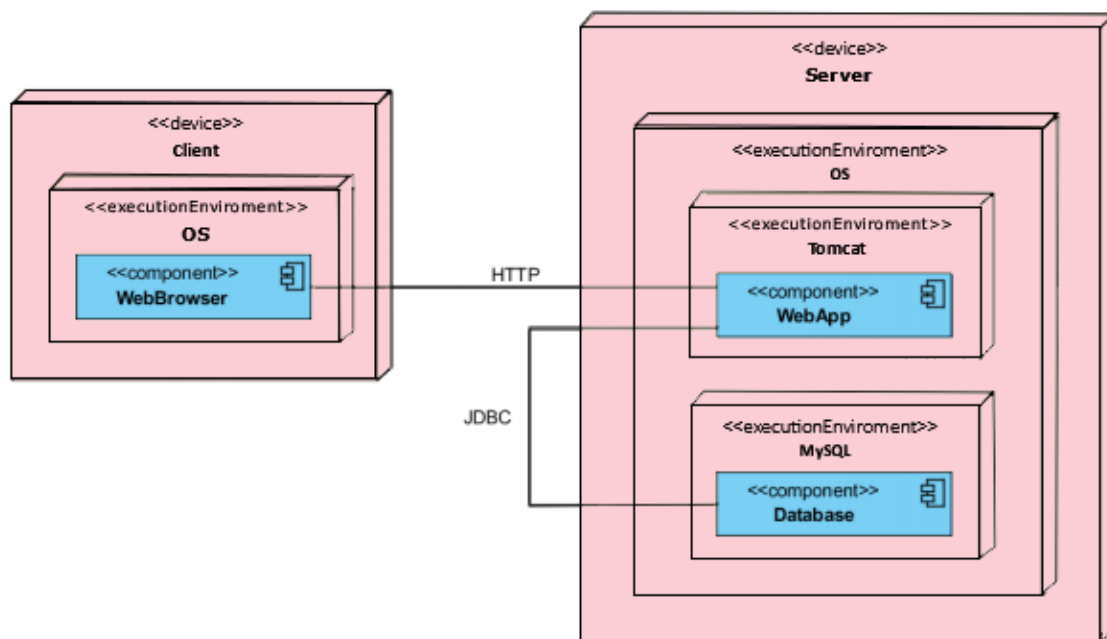
il livello di persistenza è invece composto dal DataAccess si occupa del salvataggio e dell'acquisizione di tutte le informazioni che vengono elaborate sul sito dal/sul database

3.2.3. Diagramma di deployment

Il nostro sistema è ideato per essere installato sul web server Tomcat e altri server dove è possibile utilizzare Java e MySQL.

Il sito dovrà interagire con un database presente sulla macchina secondo l'architettura client-server dove verranno gestiti i dati dei prodotti, degli ordini e degli stessi profili utente.

Alla webapp si potrà accedere tramite semplici browser preinstallati sui dispositivi più comuni.







3.3. Mapping hardware e software








Il sistema non richiede hardware particolari, infatti la proposta è costituita da un server centrale e da postazioni utenti come computer, laptop o dispositivi mobili e sarà accessibile con un qualsiasi sistema operativo preinstallato.






Il lato server verrà gestito con applicazioni come Apache Tomcat e MySQL che permetteranno a quest'ultimo di gestire i dati e mostrarli nella webapp. Tramite account specifici memorizzati nel database, i gestori del sito potranno accedere a pagine e informazioni non visibili ai semplici clienti che potranno invece soltanto navigare nella webapp ed effettuare determinate operazioni. Il database conterrà sia i dati relativi ai vari account, sia quelli relativi ai prodotti e agli ordini effettuati.

I vari client e il server verranno messi a contatto tramite una rete utilizzando il protocollo TCP/IP.






3.4. Gestione dati persistenti






| Cart | | |
|---|-------------|---------------|
|  | ID_cart | varchar(10) |
|  | ID_product | varchar(10) N |
|  | total_item | integer(10) N |
|  | total_price | float(10) N |

| Product | | |
|---|-------------|----------------|
|  | ID_product | varchar(10) |
|  | description | varchar(255) N |
|  | price | float(10) N |
|  | category | varchar(255) N |
|  | quantity | integer(10) N |
|  | discount | float(10) N |
|  | ID_cart | varchar(10) N |
|  | ID_key | integer(10) |

| Client | | |
|---|-----------|----------------|
|  | ID_client | varchar(10) |
|  | name | varchar(255) N |
|  | surname | varchar(255) N |
|  | email | varchar(255) N |
|  | password | varchar(255) N |

| Order | | |
|---|-------------|----------------|
|  | ID_order | varchar(255) |
|  | total_price | float(10) |
|  | total_item | integer(10) N |
|  | ID_cart | varchar(255) N |
|  | ID_client | varchar(255) N |

| Saved_Order | | |
|---|----------------|---------------|
|  | ID_saved_order | varchar(10) |
|  | tot_price | float(10) |
|  | ID_client | varchar(10) |
|  | quantity | integer(10) N |
|  | ID_key | varchar(10) |

| key | | |
|---|----------------|----------------|
|  | ID_key | varchar(10) |
|  | status | varchar(255) N |
|  | quantity | integer(10) N |
|  | ID_saved_order | varchar(10) N |
|  | ID_product | varchar(10) N |

3.5. Controllo degli accessi e sicurezza

Il controllo degli accessi è garantito tramite l'utilizzo di username e password per gli utenti del sistema che hanno possibilità di creare o modificare gli oggetti che modellano entità di dominio, così da prevenire accessi non autorizzati ad informazioni sensibili. Nelle prime versioni non sarà inoltre utilizzato SSL su connessione HTTP tra client e server ma non ne è escluso l'utilizzo in futuro.

3.6. Controllo flusso globale del sistema

Grazie al web container Tomcat può essere gestita l'interazione concorrente tra WebApp e più client, attraverso l'intercettazione di eventi generali proprio da questi ultimi. Tutto questo permette un controllo del flusso globale di tipo thread-driven.

In questo modo il sistema può rispondere a più utenti contemporaneamente, ma richiede che gli accessi in scrittura ai dati persistenti avvengano sequenzialmente, gestendo opportunamente le sezioni critiche.

In generale, ogni richiesta da parte di un utente verrà eseguita in un thread dedicato.

3.7. Condizioni limite

| Caso d'uso | <u>Configurazione</u> | | |
|-----------------------|--|---|---|
| Attori partecipanti | Admin | | |
| Condizioni di entrata | Il web server e il DMBS sono avviati e resi disponibili direttamente alla rete locale. | | |
| Flusso di eventi | | Admin | Sistema |
| | 1 | L'utente si reca all'interno della directory principale del server e tramite terminale accede attraverso il protocollo sftp " | |
| | 2 | | Il sistema avvia il webserver che rende disponibili le pagine web. Il webserver avvia anche il DBMS |
| Condizioni di uscita | Il sistema è avviato e pronto all' uso. | | |

| Caso d'uso | <u>Terminazione</u> | | |
|-----------------------|---|--|--|
| Attori partecipanti | Admin | | |
| Condizioni di entrata | Il web server è attualmente già avviato e in funzionate | | |
| Flusso di eventi | | Admin | Sistema |
| | 1 | L'admin accede tramite console alla directory dove è installato il web server e da terminale esegue lo spegnimento'' | |
| | 2 | | Il web server interrompe la connessione con tutti i client sulla rete |
| | 3 | | Il sistema comunica ai clienti ancora connessi l'interruzione della connessione con il server. |
| Condizioni di uscita | Il sistema non è più accessibile sulla rete locale | | |

3.7.1 Fallimento

DigiWorld può subire tre casi di fallimento, riguardanti sia l'hardware che il software

1. Fallimento dell'Hardware:
 - In questo caso il malfunzionamento può prevedere un crash del disco su cui i dati persistenti sono salvati dato che il sistema non prevede una strategia di backup e di ripristino dati.
2. Fallimento di Software:
 - Se c'è un'impossibilità di stabilire una connessione col database, il sistema mostra all'utente una schermata che sottolinea il rilevamento di un errore interno.
3. Fallimento nell'ambiente di esecuzione:
 - In caso di interruzione della fornitura elettrica, il sistema non prevede una strategia

capace di garantire l'operabilità in questo tipo di condizione.

4. Servizi dei Sottosistemi

| Utenti | |
|--------------------|--|
| Servizio | Descrizione |
| Login | Con questo sottosistema un utente può autenticarsi tramite username e password. |
| Logout | Con questo sottosistema l'utente può uscire dal sistema dopo la precedente autenticazione. |
| Utente autenticato | Con questo sottosistema l'utente ottiene i permessi derivanti dall'autenticazione. |

| Data Access | |
|--------------------------------------|--|
| Servizio | Descrizione |
| Caricamento Key Videogioco | Il sottosistema permette di ottenere dal database tutte le key dei videogiochi presenti nel sistema. |
| Caricamento Key Musica | Il sottosistema permette di ottenere dal database tutte le key delle canzoni presenti nel sistema. |
| Caricamento Key Libri | Il sottosistema permette di ottenere dal database tutte le key dei libri presenti nel sistema. |
| Caricamento Key Giftcard | Il sottosistema permette di ottenere dal database tutte le key delle giftcard presenti nel sistema. |
| Caricamento Utente | Il sottosistema permette di ottenere un nuovo utente nel database a partire dalle sue credenziali. |
| Esistenza utente | Il sottosistema permette di determinare se un utente è già presente nel database a partire dal suo username o email. |
| Caricamento richiesta d'iscrizione | Il sottosistema permette di ottenere dal database una specifica richiesta d'iscrizione. |
| Salvataggio richiesta d'iscrizione | Il sottosistema permette di scaricare su database una richiesta d'iscrizione. |
| Aggiornamento richiesta d'iscrizione | Il sottosistema permette di aggiornare su database una richiesta d'iscrizione. |

| Acquisto | |
|----------|-------------|
| Servizio | Descrizione |

| | |
|-----------------------------------|--|
| Acquisto prodotto | Il sottosistema permette di ottenere l'acquisto di un prodotto presente nel database. |
| Gestione pagamento | Il sottosistema permette di effettuare il pagamento (tramite PayPal) dei prodotti presenti nel database. |
| Elenco prodotti acquistati | Il sottosistema permette di visionare l'elenco dei prodotti acquistati presenti nel database. |

| | |
|---------------------------------------|---|
| Carrello | |
| Servizio | Descrizione |
| Aggiunta prodotto carrello | Il sottosistema permette di aggiungere prodotti nel carrello. |
| Modifica prodotto nel carrello | Il sottosistema permette di modificare le quantità di un prodotto nel carrello. |
| Rimuovi prodotto carrello | Il sottosistema permette di rimuovere prodotti nel carrello. |
| Checkout | Il sottosistema che rimanda all'acquisto dei prodotti nel carrello. |

| | |
|--|--|
| Prodotto | |
| Servizio | Descrizione |
| Visualizzazione immagini del prodotto | Il sottosistema permette di visualizzare le immagini del prodotto come anteprima. |
| Visualizzazione descrizione prodotto | Il sottosistema permette di inserire una descrizione del prodotto, e le sue caratteristiche. |
| Visualizzazione commenti | Il sottosistema permette di visualizzare i commenti lasciati dagli utenti sul prodotto. |
| Scrittura commenti | Il sottosistema permette di far scrivere commenti agli utenti riguardanti il prodotto. |

| | |
|--|---|
| Addetto servizio clienti | |
| Servizio | Descrizione |
| Aggiunta richiesta di supporto | Il sottosistema permette di aggiungere una richiesta di supporto al servizio clienti. |
| Archiviazione richiesta di supporto | Il sottosistema permette di archiviare una richiesta di supporto al servizio clienti. |
| Elenco richieste di supporto | Il sottosistema permette di elencare le richieste di supporto al servizio clienti. |
| Ottenimento richiesta di supporto | Il sottosistema permette di ottenere tutte le informazioni relative alla richiesta di supporto al servizio clienti. |

| | |
|--|--|
| Gestore Catalogo | |
| Servizio | Descrizione |
| Aggiunta di un nuovo prodotto al catalogo | Il sistema permette, al gestore, di aggiungere un nuovo prodotto al catalogo. |
| Gestione prodotto del catalogo | Il sistema permette, al gestore, di gestire il prodotto presente nel catalogo. |
| Rimozione prodotto dal catalogo | Il sistema permette, al gestore, di rimuovere un prodotto presente nel catalogo. |

Glossario

Java: linguaggio di programmazione orientato agli oggetti

Applicazione web: programma accessibile tramite browser web ed in grado di elaborare richieste e risposte HTTP

Storage: memoria persistente

Throughput: misura della capacità del sistema di condurre task contemporanei

Web Server: macchina connessa alla rete dotata di un ambiente di esecuzione

Application Server: sistema software per la gestione delle richieste/risposte provenienti dai client

Tomcat: specifico application server

DBMS: sistema software per la gestione dei dati persistenti su database

MySQL: specifico DBMS