



# Documentazione System Design

## “GLITCH”

**Progetto presentato da:**

Annunziata Elefante

Ferdinando Napolitano

Santolo Mutone

# INDICE

<b>1. Introduzione</b>	<b>3</b>
1.1. Scopo del sistema	3
1.2. Obiettivi di progettazione	3
1.3. Definizioni, acronimi e abbreviazioni	3
1.4. Panoramica	4
<b>2. Architettura software corrente</b>	<b>5</b>
<b>3. Architettura software proposta</b>	<b>5</b>
3.1. Panoramica	5
3.2. Decomposizione del sottosistema	6
3.3. Mappatura hardware/software	7
3.4. Gestione dei dati persistenti	8
3.5. Controllo e sicurezza degli accessi.	9
3.6. Controllo globale del software	10
3.7. Condizioni limite	11
<b>4. Servizi del sottosistema</b>	<b>12</b>
<b>5. Glossario</b>	<b>13</b>

# 1. Introduzione

## 1.1. Scopo del sistema

Il progetto Glitch è stato commissionato con le specifiche di permettere a tutti gli utenti registrati di poter acquistare, comodamente da casa, tutti i videogiochi e le console che desiderano.

Il sistema proposto andrà a velocizzare la vendita dei prodotti sopra citati, permettendo il passaggio da un negozio fisico, in cui il cliente deve recarsi senza avere la certezza di trovare ciò di cui necessita, a un e-commerce ricco di prodotti e di facile utilizzo. Ne segue che il software sarà rivolto a tutti gli appassionati di “gaming”, ai quali sarà consentito di inserire i prodotti scelti in un apposito carrello personale e acquistarli comodamente dal proprio divano.

Tenendo presente le specifiche, Glitch dovrà permettere di filtrare i prodotti mediante appositi filtri, consentirne l’acquisto o la rimozione dal carrello dei prodotti, permettere un servizio di assistenza, gestire il catalogo.

## 1.2. Obiettivi di progettazione

Glitch è stato sviluppato per semplificare le procedure di visualizzazione e vendita di console e videogiochi. Ciò grazie ad interfacce intuitive che consentono di interagire e gestire il sistema, garantendo all’utente una maggior *usabilità*.

Il sistema si appoggia su un DBMS esterno, sfruttandone al massimo le potenzialità di ottimizzazione per il recupero dei dati. La documentazione gestita da Glitch è a carattere riservato e ad uso esclusivo dell’utente, pertanto si è scelto di adottare un sistema di autenticazione che controlla l’accesso al sistema tramite l’inserimento di un username e una password assegnati agli utenti al momento della registrazione, ciò per garantirne l’*affidabilità*.

Il linguaggio di programmazione scelto per il suo sviluppo è Java, con paradigma di programmazione ad oggetti, che consente l’inserimento di commenti che ne semplificheranno la *sostenibilità* in caso di modifiche future.

## 1.3. Definizioni, acronimi e abbreviazioni

- **SDD:** System Design Document
- **DBMS:** DataBase Management System

- **DB:** DataBase
- **GREENFIELD ENGINEERING:** Tipologia di sviluppo che comincia da zero, non esiste nessun sistema a priori e i requisiti sono ottenuti dall'utente finale e dal cliente. Nasce, perciò, a partire dai bisogni dell'utente.

## 1.4. Panoramica

Il seguente documento SDD è diviso in sezioni ed ha la seguente composizione:

- Sezione di INTRODUZIONE: vi è presente una descrizione dell'esigenza di progettare il sistema nel miglior modo per poter facilitare le operazioni degli utenti, seguita dal contesto di utilizzo del sistema per poi presentarne gli obiettivi principali. Importante è anche la presenza di un elenco di definizioni, acronimi e abbreviazioni usato per facilitare la comprensione dei concetti citati al lettore.
- Sezione sull' ARCHITETTURA SOFTWARE CORRENTE: mostra le tecnologie e le idee di progettazione prese in considerazione per lo sviluppo di Glitch.
- Sezione sull' ARCHITETTURA SOFTWARE PROPOSTA: in primo luogo si mostra una panoramica sull'idea di base di come il sistema dovrebbe essere progettato. Ne segue la presentazione dei vari sottosistemi in cui il sistema è stato decomposto. Si continua con la descrizione del mapping effettuato sia sulle componenti hardware sia sulle componenti software, seguito dal modo in cui sono stati gestiti i dati persistenti. Le altre due sessioni successive spiegano il controllo degli accessi e della sicurezza, nonché quello globale del software. Infine, vi è una presentazione delle condizioni di confine esterne al sistema.
- Sezione sui SERVIZI DEL SOTTOSISTEMA: descrive i servizi forniti da ciascun sistema in termini di operazioni.
- Sezione di GLOSSARIO: in conclusione del SDD per specificare i termini utilizzati nel documento così da evitare ambiguità.

## 2. Architettura software corrente

Nella progettazione del sistema i progettisti non si sono potuti avvalere di un sistema preesistente, in quanto il cliente non disponeva di alcun sistema di gestione e vendita prodotti on-line.

Quindi si è proceduto con un'implementazione di tipo "Greenfield Engineering", ovvero un'implementazione da zero volta a soddisfare le esigenze del cliente, prendendo anche spunto da siti simili che aiuteranno a implementare e migliorare Glitch cercando di tenere sempre presenti i requisiti.

I siti considerati come suggerimenti si possono individuare ai seguenti link: [www.amazon.it](http://www.amazon.it), esempio di sito di e-commerce generico, e [www.gamestop.it](http://www.gamestop.it), come sito di vendita console e videogiochi.

## 3. Architettura software proposta

### 3.1. Panoramica

Il sistema proposto è un software costituito da un nucleo centrale, contenente tutti i dati relativi agli utenti registrati, ai prodotti di cui il sito dispone, alle e-mail di assistenza, alle offerte proposte e al carrello utente con i relativi ordini effettuati.

Gli utenti interagiscono con il sistema centrale da diverse postazioni client previste da Glitch.

I client non hanno alcuna possibilità di interagire tra di loro, pertanto, tutte le richieste dovranno sempre essere fatte al server centrale. Ne deriva che l'architettura più idonea al progetto è di tipo Client-Server.

Le varie tipologie di utenti previste sono:

- l'**utente** classico, che si registra al sito e può effettuare vari acquisti;
- il **gestore account**, che ha il compito di definire i ruoli di alcuni utenti rendendoli gestori delle varie funzionalità del sito (del catalogo, account o assistenza);
- il **gestore catalogo**, che è colui che si occupa di inserire/rimuovere i prodotti e le offerte in modo da tenere il catalogo sempre aggiornato;
- il **gestore assistenza**, che si occupa di rispondere alle e-mail di assistenza degli utenti in modo tale da poter aiutare e risolvere i vari problemi che gli utenti possono riscontrare.

### 3.2. Decomposizione del sottosistema

Considerando i requisiti individuati per Glitch, il sistema è stato decomposto in una serie di sottolivelli raggruppabili in 3 livelli che si occupano di gestirne aspetti e funzionalità differenti:

1. **Presentation**: raccoglie e gestisce l'interfaccia grafica e gli eventi generati dall'utente;
2. **Application**: si occupa della gestione della logica del sistema;
3. **Storage**: si occupa della gestione e dello scambio dei dati tra i sottosistemi.

All'interno del *livello Presentation* sono stati individuati i seguenti sottolivelli:

- *"InterfacciaUtente"* (che comprende l'interfaccia *"Catalogo"*) che interagisce con i sottosistemi *"Carrello"*, *"AssistenzaClienti"* e *"PaginaPersonale"*. A loro volta le interfacce *"Carrello"* e *"PaginaPersonale"* si interfacciano rispettivamente con *"Pagamento"* e *"ModificaProfilo"*;
- *"InterfacciaGestoreAccount"*, che interagisce con il sottosistema *"GestioneRuoli"*;
- *"InterfacciaGestoreCatalogo"*, che interagisce con i sottosistemi *"GestioneProdotti"* e *"GestioneOfferte"*;
- *"InterfacciaGestoreAssistenza"*, che interagisce con il sottosistema *"GestioneAssistenza"*.

Nel *livello Application* vi sono i seguenti sottolivelli:

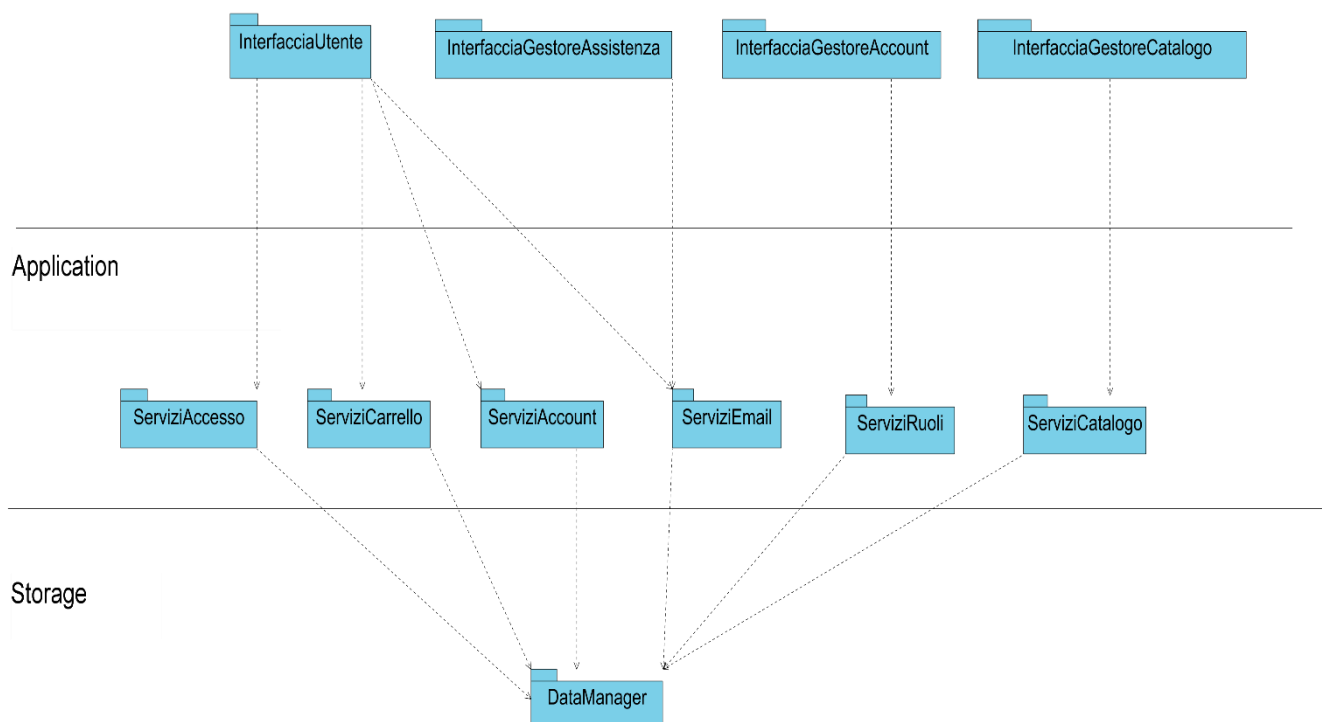
- *"ServiziAccesso"*, che si occupa di validare i dati in caso di registrazione, di gestirne le operazioni connesse e autenticare l'utente informando il Client su che tipo di interfaccia principale deve mostrare in base ai dati inseriti;
- *"ServiziAccount"*, che si occupa di gestire tutte le operazioni legate alla modifica del profilo utente e alla sua rimozione;
- *"ServiziRuoli"*, che gestisce l'assegnazione o l'eliminazione di un ruolo ad un utente;
- *"ServiziCarrello"*, che gestisce tutte le operazioni collegate al carrello da parte del Client, nonché la gestione del sottosistema *"Pagamento"*;
- *"ServiziCatalogo"*, che gestisce tutte le validazioni e le operazioni collegate all'inserimento e alla rimozione di prodotti e offerte all'interno della *"vetrina virtuale"* del sistema;
- *"ServiziEmail"*, che gestisce l'invio di e-mail di assistenza e le relative risposte.

Nel *livello Storage* vi è un unico sottosistema:

- *"DataManager"*, che si occupa di interfacciare le richieste effettuate dai vari sottosistemi al sottosistema Database.

### 3.2.1. Decomposizione in sottosistemi

Presentation

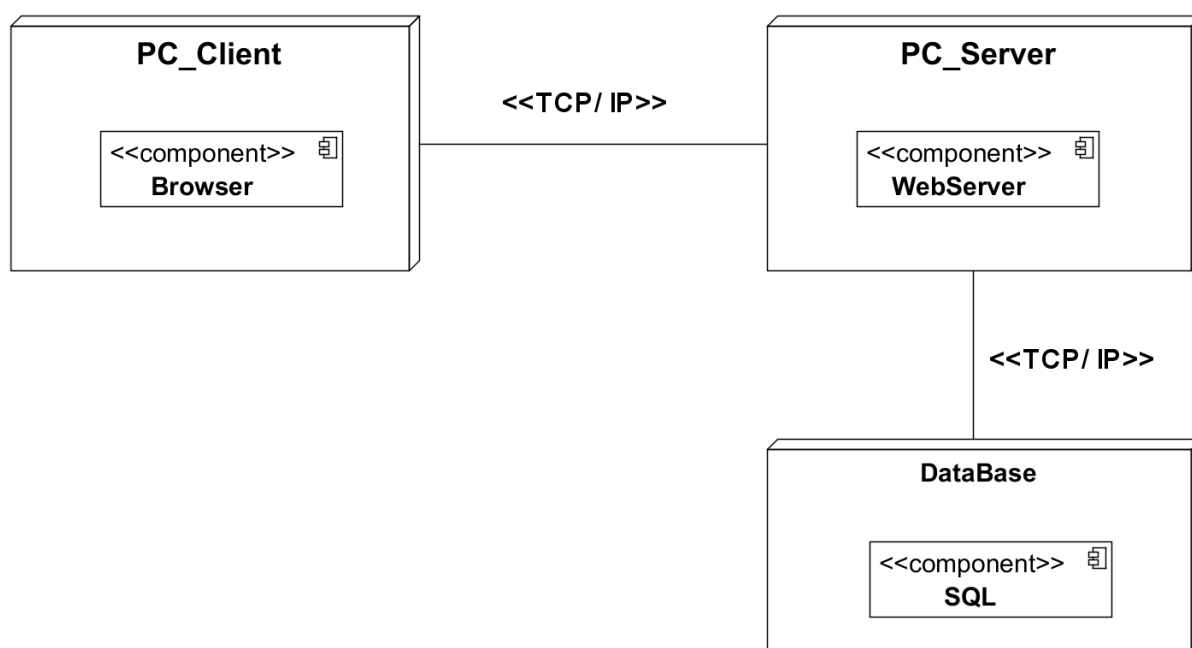


### 3.3. Mappatura hardware/software

La struttura hardware proposta è di tipo **Client-Server**, costituita da un server centrale (che presenta una struttura MVC) e da client utenti che possono essere costituiti da computer portatili o fissi nei quali possono girare Browser diversi fra loro. Al server si collegano i client ed il database, il tipo di funzionalità del client è determinata in fase di autenticazione controllando nel DB il tipo di utente che corrisponde allo username inserito in quel client. Gli utenti dovranno effettuare richieste al server per ottenere le pagine web ed eseguire le operazioni legate alle loro rispettive funzionalità. Il database conterrà i dati relativi agli account, ai prodotti, alle offerte e alle e-mail. Il client e il server saranno connessi tramite una rete che utilizzerà il protocollo TCP/IP.

In questo modo i tre livelli *Presentation (View)*, *Application (Controller)* e *Storage (Model)* vengono tutti integrati all'interno del lato server.

### 3.3.1. Diagramma di Deployment



## 3.4. Gestione dei dati persistenti

Il sistema Glitch si avvale dell'uso di un database di tipo relazionale, al quale si interfaccia tramite un DB. Questa scelta è stata ponderata data la sicurezza offerta da un DB di ultima generazione, assieme ad una maggiore affidabilità e garanzia di coerenza e facilità di gestione, nonché dalla velocità di accesso e trasmissione dei dati.

### 3.4.1. Tabella priorità dei dati

Nome indicatore	Dati	Priorità
Credenziali	<ul style="list-style-type: none"><li>username</li><li>password</li></ul>	Urgente
Dati personali	<ul style="list-style-type: none"><li>nome</li><li>cognome</li><li>e-mail</li><li>indirizzo</li><li>ruolo</li></ul>	Urgente
Dati prodotti e ordini	<ul style="list-style-type: none"><li>nome</li><li>genere</li><li>piattaforma</li></ul>	Urgente



	<ul style="list-style-type: none"> <li>• modello</li> <li>• casa produttrice</li> <li>• immagine</li> <li>• prezzo</li> </ul>	
<b>Dati offerta</b>	<ul style="list-style-type: none"> <li>• nome</li> <li>• percentuale</li> <li>• categoria</li> </ul>	Urgente
<b>Dati pagamento</b>	<ul style="list-style-type: none"> <li>• intestatario (nome e cognome)</li> <li>• numero identificativo</li> <li>• scadenza</li> <li>• CVV</li> </ul>	Consigliata
<b>Dati assistenza</b>	<ul style="list-style-type: none"> <li>• destinatario</li> <li>• oggetto</li> <li>• descrizione</li> <li>• stato</li> </ul>	Urgente

### 3.5. Controllo e sicurezza degli accessi

La sicurezza ed il controllo dei dati del database è garantita mediante l'utilizzo di credenziali, costituite da username e password, per ogni singolo utente, che verranno richieste per ogni singolo accesso. Ogni tipo di utente avrà a disposizione diverse interfacce grafiche (nel caso il ruolo lo richieda), in modo che ogni utente possa accedere solo alle funzionalità che rientrano nella sua categoria di utenza.

#### 3.5.1. Tabella accessi

Sottosistema Attore	Gestione operazione					
	Accesso	Account	Acquisto	Prodotti	Ruoli	Assistenza
Utente	R, Li, Lo	MP, RP	AP			RicA
Gestore account	R, Li, Lo	MP, RP	AP		AR, ER	
Gestore catalogo	R, Li, Lo	MP, RP	AP	IP, RP, IO, RO		
Gestore assistenza	R, Li, Lo	MP, RP	AP			RisA

#### Legenda:

- **R** = Registrazione
- **Li – Lo** = Login – Logout
- **MP** = Modifica profilo
- **RP** = Rimozione profilo
- **AR** = Aggiungi Ruolo
- **ER** = Elimina ruolo
- **AP** = Acquisto prodotto  
(comprende tutte le operazioni  
annesse al carrello)
- **IP** = Inserimento prodotto
- **RP** = Rimozione prodotto
- **RicA** = Richiesta assistenza
- **RisA** = Risposta assistenza
- **IO** = Inserimento offerta
- **RO** = Rimozione offerta

### 3.6. Controllo globale del software

Il sistema è caratterizzato da un portale accessibile da **Browser** e da un **WebServer** attivo 24/7, che deve provvedere a gestire gli accessi concorrenti da parte dei vari utenti. Quando un utente si logga e sottomette i propri dati, vi è un accesso al DB (query di interrogazione) che permette di controllare l'esistenza del soggetto.

Dopo la conferma, l'utente può accedere a diverse operazioni messe a disposizione dal sistema in base al suo ruolo. L'utente base può accedere al sistema, visualizzare il catalogo, acquistare prodotti e modificare/rimuovere il proprio profilo.

I gestori, in aggiunta alle operazioni precedenti, gestiscono delle funzionalità di Glitch in base ai propri ruoli (chi gestisce gli account, chi il catalogo e chi l'assistenza).

Nella maggior parte delle operazioni effettuate dagli utenti si accede al DB in scrittura (si pensi alla registrazione o all'inserimento prodotto), mentre per le semplici visualizzazioni di dati ciò non è necessario.

L'intero sistema presente una *struttura centralizzata*, in quanto tutti i dati vengono gestiti all'interno di una singola componente: il server.

## **3.7. Condizioni limite**

### **3.7.1. Inizializzazione**

Per l'inizializzazione del sistema Glitch è necessario l'avvio di un web server che fornisca il servizio di un Database per la gestione dei dati persistenti e l'interpretazione ed esecuzione del codice lato server. In seguito, tramite l'interfaccia di Login, sarà possibile autenticarsi tramite opportune credenziali (username e password) come utente che, in base al suo ruolo, ha pieno accesso a tutte le funzionalità del sistema.

Una volta effettuato l'accesso, Glitch presenterà all'utente la sua homepage, dalla quale si possono effettuare tutte le operazioni che il sistema fornisce.

### **3.7.2. Terminazione**

Al momento della chiusura del sito si ha la terminazione del sistema con una regolare Logout. Viene assicurata la consistenza dei dati, annullando eventuali operazioni che erano in esecuzione.

### **3.7.3. Fallimento**

Il sistema lato server può fallire solo a causa di condizioni eccezionali quali mancanza di elettricità o guasti all'hardware (hard disk danneggiato ecc., che danneggiano permanentemente il sistema) o in caso di crash di sistema (attacchi al server dall'esterno). Il sistema lato client può fallire a causa di guasti temporanei (hardware o software), o per la caduta della linea telefonica, ma ciò non influisce in alcun modo sul lato server (che resterà comunque stabile), neanche se ci fosse stata una connessione attiva al DB, visto che il DBMS gestisce le transazioni. Per recuperare da un fallimento basta riaccendere il web server, ma ovviamente, nel frattempo, i client non possono usufruire nel sistema visto che il server è down.

### **3.7.4. Eccezioni**

Il sistema lato client può lanciare delle eccezioni mirate in caso di operazioni errate dell'utente (login errato) o non consentitegli (accedere al carrello se non si è registrati), in questo modo non vi è un crash della pagina.

## 4. Servizi del sottosistema

Il *livello Presentation* è suddiviso in una serie di sottosistemi che gestiscono le interfacce visibili da un utente, sia esso un cliente o un gestore con uno specifico ruolo, e gli eventi generati dalla sua interazione con il sistema.

I *livelli Application e Storage* sono costituiti da una serie di sottosistemi che gestiscono tutte le richieste di servizi lato Client, rappresentato da Presentation. Abbiamo:

- **ServiziAccesso** offre 3 servizi all'*InterfacciaUtente*:
  - Registrazione -> risponde a richieste di registrazione al sito da un utente, se non è già stato registrato in precedenza;
  - Login -> risponde a richieste di un utente di Glitch di accedere al sito;
  - Logout -> risponde a richieste di un utente di Glitch di lasciare il sito.
- **ServiziAccount** offre 2 servizi alla *PaginaPersonale*:
  - Modifica profilo -> risponde a richieste di un utente di modificare alcuni dati personali: e-mail, indirizzo e password;
  - Rimozione profilo -> risponde a richieste di un utente di eliminare il proprio profilo dal sito.
- **ServiziCarrello** offre 3 servizi al *Carrello*:
  - Aggiungi prodotto al carrello -> risponde a richieste di un utente di inserire un prodotto all'interno del proprio carrello;
  - Rimozione prodotto dal carrello -> risponde a richieste di un utente di rimuovere un prodotto all'interno del proprio carrello;
  - Acquisto prodotto -> risponde a richieste di un utente di acquistare i prodotti contenuti all'interno del proprio carrello.
- **ServiziEmail** offre 2 servizi alle interfacce *AssistenzaClienti* e *GestioneAssistenza*:
  - Richiesta assistenza -> risponde a richieste di un utente di inviare un'e-mail ai gestori per richiedere assistenza in caso di problemi;
  - Risposta assistenza -> risponde a richieste del gestore assistenza di rispondere a eventuali e-mail.
- **ServiziRuoli** offre 2 servizi a *GestioneRuoli*:
  - Aggiungi ruolo -> risponde a richieste del gestore account di assegnare un ruolo ad un utente del sito;
  - Elimina ruolo -> risponde a richieste del gestore account di eliminare un ruolo ad un utente del sito.
- **ServiziCatalogo** offre 4 servizi alle interfacce *GestioneProdotti* e *GestioneOfferte*:
  - Inserimento prodotto -> risponde a richieste del gestore catalogo di inserire un nuovo prodotto all'interno del catalogo del sito;
  - Rimozione prodotto -> risponde a richieste del gestore catalogo di eliminare un prodotto dal catalogo del sito;
  - Inserimento offerta -> risponde a richieste del gestore catalogo di aggiungere un'offerta su alcuni dei prodotti del catalogo del sito;
  - Rimozione offerta -> risponde a richieste del gestore catalogo di eliminare un'offerta su alcuni dei prodotti del catalogo del sito.

- **ManagerDBMS** offre servizi di inserimento/modifica/rimozioni/prelievo dei dati su DBMS alle richieste provenienti dai Control.

## 5. Glossario

**Browser:** programma per scorrere velocemente una banca dati, in particolare per la navigazione su Internet.

**WebServer:** un'applicazione software che, in esecuzione su un server, è in grado di gestire le richieste di trasferimento di pagine web di un client, ovverosia di un browser.

**Database:** i database sono sistemi di gestione dei dati elettronici logicamente strutturati, che con l'aiuto di un sistema di gestione del database regolano appartenenze e diritti di accesso e archiviano le informazioni riguardo alla base di dati contenuta. La maggior parte dei database si possono aprire, modificare e leggere soltanto con particolari applicazioni per database. Tra i software più diffusi per la gestione di database è annoverato **MySQL**.