

GIOVANNI PELLEGRINI



CONCETTI FONDAMENTALI DI WEB APPLICATION

TECNICO SUPERIORE PER L'AUTOMAZIONE ED I SISTEMI MECCATRONICI

Profilo : HIGHER TECHNICIAN BARI 4.0 (HTECH 6)

2024

Dispensa didattica



Area nuove tecnologie per il Made in Italy
Sistema meccanico/meccatronico

INDICE

1. Definizione di Web Application	pag. 3
2. Breve sintesi storica delle Web Applications	
3. Caratteristiche delle Web Applications	pag. 4
4. Esempi e ambiti di sviluppo delle Web Applications	pag. 6
5. Glossario ragionato	pag. 8
6. Sitografia	pag. 15

1. Definizione di Web Application

In informatica l'espressione **applicazione web**, (ovvero web application in inglese), indica genericamente tutte le applicazioni distribuite web-based.

Nell'ingegneria del software e nella programmazione web essa indica, infatti, un'applicazione accessibile/fruibile, via web, per mezzo di un network, come ad esempio una Intranet all'interno di un sistema informatico o attraverso la Rete Internet, ovvero in una architettura tipica di tipo client-server, che offre determinati servizi all'utente client. Un generico client accede connettendosi a funzionalità applicative residenti su un application server utilizzando, come terminali utente, normali web browser, appoggiandosi, dunque, ai consueti protocolli di rete. I client finalizzati unicamente a collegarsi come terminali di web application vengono chiamati sovente thin client.

Questo modello applicativo è divenuto piuttosto popolare alla fine degli anni Novanta con la diffusione di Internet, in un contesto di Web 2.0 ovvero web dinamico. Infatti, l'opportunità di aggiornare ed evolvere a costo ridotto il proprio applicativo, senza essere costretti a distribuire numerosi aggiornamenti ai propri clienti attraverso supporti fisici, ha reso la soluzione piuttosto popolare per molti produttori software.

Più di recente società come Google e Microsoft hanno implementato interi pacchetti applicativi per office automation, tradizionalmente venduti in modo distribuito su supporti CD-ROM, e che ora si stanno velocemente trasformando a tutti gli effetti in webapps (es. Google Docs). Spesso sono applicazioni web complesse di tipo enterprise, ovvero di business ad uso interno prettamente aziendale o indirizzati ai propri clienti esterni.

2. Breve sintesi storica delle Web Applications

Quando le prime applicazioni client-server cominciarono ad essere diffuse nel panorama informatico, da metà circa degli anni Ottanta, questi sistemi richiedevano che per ognuno di essi fosse installata la parte client direttamente sulla workstation di ciascun utente. In breve qualunque aggiornamento, patch o modifica dell'applicativo lato server (server-side) si trasformava nella esigenza di installare le versioni aggiornate su tutte le workstation degli utenti dell'applicativo stesso. Questo

tipo di problematiche inducevano ovviamente, in modo diretto, un onere economico non indifferente per il produttore del software, e di conseguenza un costo periodico di adeguamento per tutti gli utenti di questo tipo di applicazione informatica.

Di contro, le web-application generano il loro output in modo dinamico pilotando un front-end composto da documenti Web, destinati quindi ad un semplice web browser.

Le prime web apps consistevano infatti nella generazione di pagine standard HTML/XHTML; successivamente con l'evolversi delle tecnologie associate e soprattutto con la nascita di nuovi standard, si cominciarono a "distribuire" attraverso di esse documenti in formati ancora più "neutri", come l'XML.

In breve ogni singola pagina Web viene inviata al client come fosse un documento statico, ma la sequenza delle pagine può fornire un'esperienza interattiva, poiché l'input dell'utente

è restituito attraverso gli elementi embedded della pagina stessa. Durante la sessione, il web browser interpreta e visualizza le pagine, e dunque diventa il client universale per tutte le web-application, indipendentemente dal produttore del software.

Più di recente attraverso le webapps vengono spesso passati flussi di dati in formati SGML generici (o addirittura custom) direttamente ad altre applicazioni che, in virtù dell'uso di questo tipo di output, sono sempre accessibili attraverso un comune web browser.

La parte dinamica lato client (o client-side) di questi sistemi informatici (es. applet) è affidata sempre a linguaggi standard, come ad esempio JavaScript, che sono inclusi in tutti i browser. Il crescente successo conseguito da librerie esperte, ormai veri e propri framework come JQuery, tecnologie dinamiche come AJAX, oppure plug-in, come il conosciutissimo Flash Player, consente oggi di "pilotare" ed arricchire le interfacce utente in modo completo ed efficiente.

3. Caratteristiche delle Web Applications

Spesso è importante che i visitatori di un sito web vedano che il contenuto sia coerente ed aggiornato. Il contenuto di un sito che varia nel tempo necessita di essere cambiato continuamente. Per esempio, in un sito web di e-commerce è richiesto che vengano pubblicati solamente gli annunci relativi a prodotti non ancora venduti. È anche importante che i nuovi annunci siano pubblicati al massimo uno o due giorni dopo che sono stati inviati dal venditore.

Se una di queste condizioni non viene rispettata, il sito probabilmente non avrà molto successo.

L'impaginazione del testo e delle immagini che compaiono nel web browser quando l'utente visita un sito web è spesso creata utilizzando un semplice linguaggio noto come **Hyper Text Markup Language (HTML)**.

Quando un utente visita un sito web, la porzione di testo che è "delimitata" dall'HTML viene trasferita dal sito web al browser dell'utente. Il browser interpreta questo testo, mostrando testo ed immagini all'utente. La porzione di testo che viene trasferita è tipicamente chiamata **pagina web**.

Molti visitatori di siti web concepiscono la navigazione in termini di spostamenti "da pagina a pagina" all'interno di un sito.

Quando fanno click con il mouse su un collegamento ipertestuale vengono trasportati dai loro browser in un'altra pagina.

Quando premono il pulsante Back invece vengono riportati all'ultima pagina che hanno visitato.

Alcuni siti web sono statici. I siti web statici richiedono una persona con un livello di accesso privilegiato (a volte definita webmaster) per "aggiornare" manualmente il contenuto.

L'aggiornamento del contenuto richiede che la persona visiti ed aggiorni manualmente l'HTML delle pagine che devono cambiare.

Generalmente, questo viene fatto modificando un insieme di file sul server web (il computer che fa "girare" il sito web), in cui ogni file rappresenta una singola pagina.

Le modifiche all'aspetto di un sito web statico richiedono che il manutentore del sito visiti ed aggiorni ogni file che compone il sito web. I siti web tipicamente possono crescere fino a comprendere migliaia di file e per questo l'operazione può non essere un compito banale. Il manutentore responsabile del sito di annunci di automobili ha l'onere aggiuntivo di tenere aggiornati anche gli annunci stessi.

Se ogni pagina nel sito web rappresenta un annuncio relativo ad una particolare automobile, il webmaster ha la necessità di eliminare le pagine che compongono l'annuncio scaduto e di creare le pagine per le nuove inserzioni.

Quindi ha anche l'esigenza che nessun collegamento su altre pagine punti alle pagine rimosse.

La quantità di lavoro da compiere diventa molta in un tempo davvero breve.

Il manutentore del sito può anche, comprensibilmente, commettere errori e dimenticarsi di aggiornare o rimuovere pagine importanti.

Da qui l'esigenza di sostituire le operazioni ripetitive e manuali con soluzioni applicative automatizzate, che svolgano queste funzioni in modo più efficiente, rapido e sicuro.

Infatti, in un sito web generato dinamicamente (detto in questo caso sito dinamico), al manutentore non è richiesto di visitare ogni pagina per eseguire l'aggiornamento del contenuto o dello stile.

Se il nostro ipotetico manutentore del sito degli annunci di automobili decidesse di costruire un'applicazione web per gestire questo sistema, potrebbe conservare una lista di "nuovi annunci" slegata dalla struttura HTML (memorizzandola ad esempio in una generica base di dati).

Sarebbe quindi in grado di istruire la propria applicazione web perché, quando un utente visita il sito, questa interroghi la base di dati e generi il codice HTML corrispondente ad uno di questi "annunci recenti" ovvero al loro indice.

Per sua natura una webapp può presentarsi con diverse strutture ed organizzazioni logiche, poiché di fatto racchiude in sé, allo stesso tempo, un modello tecnico ed una filosofia di sviluppo.

Tuttavia, sul piano dell'informatica teorica è possibile riconoscere una strutturazione tipica su più livelli (architettura multi-tier).

Nella maggioranza dei casi è infatti possibile identificare:

- logica di presentazione (o presentation layer): primo livello associabile al terminale di fruizione, visualizzazione o presentazione a favore dell'utente (front-end) attraverso il motore di rendering del web browser del client ovvero l'interfaccia utente (tipicamente realizzata tramite pagine HTML e CSS, JSP o JSF in tecnologia Java EE, JavaScript).
- logica di business (o business logic): secondo livello costituito dal motore applicativo, ovvero un core applicativo (back-end) o logica applicativa o di elaborazione presente tipicamente su un application server e costituita da codice sorgente in un qualche linguaggio di sviluppo dinamico lato-server (per es. PHP, ASP); questa riceve, elabora e soddisfa le richieste del client.

La presenza di questo strato è ciò che differenzia un'applicazione web da un normale sito web, in quanto implementa il servizio web sotto forma di codice sorgente che essa fornisce al client.

- strato dati (o data layer): terzo eventuale livello riconducibile al motore database associato (per es. MySQL, MSSql, Oracle, DB2,) per la gestione della persistenza dei dati e la loro interrogazione, ricevendo e soddisfacendo le richieste di lettura/scrittura sul DB da parte della logica applicativa. Può risiedere sull'application server assieme alla logica applicativa o direttamente sul DB.

Ricapitolando, il web browser del client invia le proprie richieste, tramite il protocollo HTTP/HTTPS, al livello intermedio ovvero al motore applicativo dinamico del web server, che da una parte interpreta e gestisce le interrogazioni al motore DBMS e dall'altra genera il risultato in un output diretto allo stesso browser, che lo interpreta e lo restituisce all'utente sotto forma di Web.

Un'applicazione web si caratterizza dunque essenzialmente per il trasferimento di dati o informazioni da uno strato all'altro ovvero dal front-end fino al back-end e viceversa con in aggiunta eventuali elaborazioni/computazione sulla parte di core business.

Spesso inoltre in applicazioni web di una certa dimensione o complessità, al pari di siti web e portali, è presente una pagina iniziale (home page) con un menù iniziale e relativi sottomenù per l'insieme di funzionalità o servizi offerti all'utente.

Come per le altre applicazioni lo sviluppo delle applicazioni web avviene tipicamente seguendo il ciclo di vita del software a partire da requisiti/specifiche.

Il lavoro del programmatore web tipicamente avviene in locale sul proprio PC con IDE installato e configurato.

A differenza delle altre forme di programmazione lo sviluppo in locale delle applicazioni web necessita del supporto di un application server installato sempre in locale sul PC dello sviluppatore che funzioni da backend su cui effettuare il deployment dell'applicazione ad ogni passo dello sviluppo testando interfaccia utente e business logic, verificando poi il risultato prodotto (front-end) digitando il relativo url sul web browser.

Spesso, sempre ai fini di test dell'applicazione è necessario installare in locale anche un database dei dati che funzioni da data layer ovvero verificandone la persistenza.

4. Esempi e ambiti di sviluppo delle Web Applications

Le applicazioni web-based oggi concorrono ad implementare in tutto o in parte soluzioni software all'interno di motori di ricerca, web forum, blog, webmail, social network, CMS, e-commerce, MMORPG (Massively Multiplayer Online Role-Playing Game, ovvero "gioco di ruolo multigiocatore in rete di massa"), gestionali di vario tipo e molto altro ancora.

Troviamo applicazioni web un po' ovunque, ma nel linguaggio comune ci si riferisce ad esse come semplici "siti web" o portali web evoluti.

Esempi didattici tipici sono:

- **bacheca di annunci di lavoro:** l'azienda registrata pubblica il suo annuncio di lavoro e l'interessato risponde inviando tramite form le sue generalità e l'upload del proprio curriculum.

- **registrazione di un utente su un forum tramite pagina iniziale di form** e successivo inserimento delle generalità utente in un database da parte della logica di elaborazione per successivo login;
- **ricerca di un sito tramite un motore di ricerca**: l'utente digita la parola chiave da cercare sull'interfaccia web sul web browser ed invia il dato ad un server che tramite opportuni algoritmi esegue una query particolare su un database per l'individuazione dell'indirizzo della risorsa web ricercata.
- **una pagina di commento di un blog o di un forum**, dove l'utente inserisce nell'interfaccia web (text area) il proprio commento, inviato poi alla logica di elaborazione per l'inserimento nel database e reso disponibile ad altri utenti visualizzatori della pagina.
- una pagina di login per l'accesso alle funzionalità e servizi di un sito, portale, forum, attraverso il form iniziale che rappresenta la parte di presentazione all'utente dell'applicazione, una parte di elaborazione che esegue il controllo dei dati acquisiti dal sistema con quelli presenti nel database per la loro validazione o meno ed infine una pagina di conferma o errore per l'operazione effettuata dall'utente;
- **post di testo, immagini o video tramite interfaccia web su un social network** che vengono inserite sul database e rese disponibili ai rispettivi contatti;
- in **un servizio di webmail** l'utente invia un'email tramite l'interfaccia web al server di posta che provvede all'inserimento del messaggio nella rispettiva casella di posta (boxmail) del database, oltre che all'invio al destinatario;
- **una piattaforma Wiki (es. Wikipedia)**: una modifica del testo di una voce viene passata ad una logica di elaborazione che provvede all'inserimento della modifica nel database; viceversa una richiesta di voce è una query sul database o repository.
- **gestione dell'anagrafica dei curriculum vitae all'interno di una società o di un motore di ricerca di lavoro** dove l'utente in fase di scrittura inserisce il CV attraverso una pagina di form lato front-end, i cui dati vengono passati prima alla parte di business per eventuali elaborazioni e poi infine memorizzati nel lato back-end di persistenza del database, e viceversa in fase inversa di lettura.
- **siti di e-commerce (es. eBay)**: l'utente acquista uno o più prodotti selezionandoli e registrandoli in memoria (sessione) nel carrello acquisti ed invoca l'ordine memorizzato in un database dopo l'esecuzione della transazione finanziaria per il pagamento dovuto fornendo i dati della propria carta di credito.

Applicazioni web più complesse sono invece quelle bancarie per la gestione di pratiche (tipicamente ad uso interno tramite Intranet) e flussi di denaro e transazioni finanziarie (es. home banking) tra istituto di credito e clienti.

Altri **esempi pratici più complessi sono ad esempio i servizi di webmail (es. Gmail), le piattaforme social network come Facebook, LinkedIn, Instagram, Flickr, i CMS WordPress, Joomla, Drupal, Microsoft Sharepoint, quelle applicazioni che ci permettono di fare ricerche sul web come Google; di collaborare a progetti, come SourceForge; di acquistare prodotti da un'asta, come avviene su eBay.** Ma questo è solo un esempio poiché usando un qualsiasi motore di ricerca è facile scoprire un'infinità di applicazioni web in tutte le lingue e paesi.

5. Glossario ragionato

Intranet

In informatica e telecomunicazioni l'**intranet** è una rete aziendale privata che utilizza il protocollo TCP/IP, ma può estendersi anche con collegamenti WAN e VPN. Spesso tale rete o è completamente isolata dalla Rete Internet esterna (es. LAN), rimanendo a solo uso interno, oppure comunica eventualmente con la rete esterna e le altre reti, attraverso opportuni sistemi di comunicazione e relativa protezione (come ad esempio un firewall).

Client - Server

In informatica il termine **sistema client-server** (letteralmente *cliente-serviente*) indica un'architettura di rete nella quale genericamente un computer *client* o terminale si connette ad un *server* per la fruizione di un certo servizio, quale ad esempio la condivisione di una certa risorsa hardware/software con altri client, appoggiandosi alla sottostante architettura protocollare.

Più semplicemente, i sistemi *client/server* sono un'evoluzione dei sistemi basati sulla condivisione semplice delle risorse: la presenza di un *server* permette ad un certo numero di *client* di dividerne le risorse, lasciando che sia il *server* a gestire gli accessi alle risorse per evitare conflitti di utilizzazione tipici dei primi sistemi informatici.

Le reti locali aziendali (LAN), la rete Internet, i sistemi informatici e i sistemi operativi sono organizzati sotto forma di una tipica architettura client-server per la fruizione dei rispettivi servizi.

Application Server

In informatica un **application server** (a volte abbreviato con la sigla **AS**) è una tipologia di server che fornisce l'infrastruttura e le funzionalità di supporto, sviluppo ed esecuzione di applicazioni nonché altri componenti server in un contesto distribuito. Si tratta di un complesso di servizi orientati alla realizzazione di applicazioni ad architettura multilivello ed enterprise, con alto grado di complessità, spesso orientate per il web (applicazioni web). La sua gestione è opera dei cosiddetti sistemisti applicativi che si occupano delle operazioni di installazione, configurazione e manutenzione dell'applicazione web.

Protocollo di rete

Per **protocollo di rete**, in telecomunicazioni, si intende un particolare tipo di protocollo di comunicazione preposto al funzionamento di una rete informatica, ovvero la definizione formale a priori delle modalità di interazione che due o più apparecchiature elettroniche,

collegate tra loro, devono rispettare per operare particolari funzionalità di elaborazione necessarie all'espletamento di un certo servizio di rete.

In termini equivalenti si tratta dunque della descrizione a livello logico del processo di comunicazione, (meccanismi, regole o schema di comunicazione) tra terminali e apparati, preposto al funzionamento efficace della comunicazione in rete. Queste apparecchiature possono essere host, computer client, smartphone, personal digital assistant (PDA), monitor, stampanti, sensori.

In particolare un protocollo prevede la definizione dei linguaggi costituiti dai messaggi scambiati, messaggi che devono potersi interpretare correttamente.

L'aderenza ai protocolli garantisce che due software in esecuzione su diverse macchine possano comunicare efficacemente, (cosiddetta interoperabilità) anche se sono stati realizzati indipendentemente.

È evidente l'importanza della standardizzazione dei protocolli di rete.

L'implementazione informatica dei protocolli di rete definisce, all'interno dell'architettura di rete, il cosiddetto software di rete, presente usualmente all'interno del sistema operativo ed elaborato dalla scheda di rete.

In senso più lato, un protocollo di comunicazione si può definire come un insieme di regole che vengono stabilite per instaurare una comunicazione corretta: ad esempio due persone di differenti madrelingue potrebbero mettersi d'accordo nell'utilizzo della lingua inglese per comunicare.

Thin client

In informatica un **thin client** (letteralmente *client sottile*) è un computer che opera come client in un sistema client/server, caratterizzato dalla presenza di nessuna o un ristretto numero di applicazioni poiché il suo esercizio dipende strettamente da un server centrale per lo svolgimento della maggior parte delle proprie funzioni.

La parola "thin" si riferisce proprio al minimalismo con cui questi client sono concepiti, e al fatto che tali computer vengono solitamente avviati tramite una piccola immagine di boot che contiene poco più del necessario per connettersi a una rete e lanciare un web browser o un desktop remoto.

Patch

In informatica una **patch** (in inglese *pezza, toppa*) indica una porzione di software progettata per aggiornare o migliorare un programma.

Ciò include la risoluzione di vulnerabilità di sicurezza e altri bug generici: tali patch vengono anche chiamati fix o bugfix. Mediante patch vengono anche migliorate l'usabilità e le prestazioni dell'applicazione.

Front end / back end

I termini **front end** (in sigla **FE**) e **back end** (in sigla **BE**) (anche scritti, con grafia meno corretta, ma più comune, *frontend* o *front-end* e *backend* o *back-end*) denotano,

rispettivamente, la parte visibile all'utente e con cui egli può interagire (interfaccia utente) e la parte che permette l'effettivo funzionamento di queste interazioni.

Il *front end*, nella sua accezione più generale, è responsabile dell'acquisizione dei dati di ingresso e della loro elaborazione con modalità conformi a specifiche predefinite e invarianti, tali da renderli utilizzabili dal *back end*.

Il collegamento del *front end* al *back end* è un caso particolare di interfaccia.

HTML / XHTML / Linguaggio di markup

In informatica l'**HyperText Markup Language (HTML)**; traduzione letterale: linguaggio a marcatori per ipertesti) è un linguaggio di markup.

Nato per la formattazione e impaginazione di documenti ipertestuali disponibili nel web 1.0, oggi è utilizzato principalmente per la struttura di una pagina web (definita appunto dal markup) e la sua rappresentazione, gestita tramite vari stili, per adattarsi alle nuove esigenze di comunicazione e pubblicazione all'interno di Internet.

L'HTML è un linguaggio di pubblico dominio, la cui sintassi è stabilita dal World Wide Web Consortium (W3C). È derivato dall'SGML, un metalinguaggio finalizzato alla definizione di linguaggi utilizzabili per la stesura di documenti destinati alla trasmissione in formato elettronico. La versione attuale, la quinta, è stata rilasciata dal W3C nell'ottobre 2014.

Il motivo principale che ha spinto il W3C e i suoi membri a sviluppare HTML5 è stata la necessità di fornire direttamente le funzionalità che in precedenza erano fruibili tramite estensioni proprietarie all'esterno dei browser, come Adobe Flash e simili.

Un secondo obiettivo che gli sviluppatori si erano prefissati era quello di garantire una maggiore compatibilità tra i diversi browser, indipendentemente dalla piattaforma software utilizzata, e principalmente mirata all'espansione dei dispositivi mobili.

L'**XHTML** (sigla di **eXtensible HyperText Markup Language**, *Linguaggio di marcatura di ipertesti estensibile*) è un linguaggio di marcatura che associa alcune proprietà dell'XML con le caratteristiche dell'HTML: un file XHTML è un pagina HTML scritta in conformità con lo standard XML.

In informatica **XML** (sigla di **eXtensible Markup Language**) è un metalinguaggio per la definizione di linguaggi di markup, ovvero un linguaggio marcatore basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo.

Costituisce il tentativo di produrre una versione semplificata di Standard Generalized Markup Language (SGML) che consente di definire nuovi linguaggi di markup.

Il **linguaggio di markup** è un insieme di regole che descrivono i meccanismi di rappresentazione (strutturali, semantici, presentazionali) di un testo; facendo uso di convenzioni rese standard, tali regole sono utilizzabili su più supporti.

Il termine *markup* (o **marcatura**) deriva dall'ambiente tipografico dove si usava contrassegnare con annotazioni le parti del testo che andavano evidenziate o corrette, allo scopo di segnalarle al compositore o al dattilografo.

Applet

In informatica il termine **applet** (neologismo degli anni novanta, composto di application e let) indica un programma che viene eseguito come "ospite" nel contesto di un altro programma, detto per questo container, su un computer client (elaborazione lato client). In altre parole, un applet è un programma progettato per essere eseguito all'interno di un programma-container; ne consegue che l'applet non può essere eseguito indipendentemente da un altro programma.

JavaScript

In informatica **JavaScript** è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client, recentemente esteso anche al lato server, per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da *eventi* innescati a loro volta in vari modi dall'utente sulla pagina web in uso (mouse, tastiera, caricamento della pagina ecc...).

Tali funzioni di script, utilizzati dunque nella *logica di presentazione*, possono essere opportunamente inserite in file HTML, in pagine JSP o in appositi file separati con estensione *.js* poi richiamati nella *logica di business*.

Ultimamente il suo campo di utilizzo è stato esteso alle cosiddette *Hybrid App* (app ibride), con le quali è possibile creare app per più sistemi operativi.

Fu originariamente sviluppato da Brendan Eich della Netscape Communications con il nome di **Mochan** e successivamente di **LiveScript**, ma in seguito è stato rinominato "JavaScript" ed è stato formalizzato con una sintassi più vicina a quella del linguaggio Java di Sun Microsystems (che nel 2010 è stata acquistata da Oracle).

JavaScript è stato standardizzato per la prima volta il 1997 dalla ECMA con il nome **ECMAScript**. L'ultimo standard è la decima edizione, chiamata ECMAScript 2019 ed è stata rilasciata nel giugno 2019.

Server web

In informatica un **server web** (o web server) è un'applicazione software che, in esecuzione su un server, è in grado di gestire le richieste di trasferimento di pagine web di un client, tipicamente un web browser.

La comunicazione tra server e client avviene tramite il protocollo HTTP, che utilizza la porta TCP 80 (o 8080), o eventualmente la versione sicura HTTPS, che utilizza invece la 443. Su un server web risiedono dunque i siti web tramite hosting. L'insieme di tutti i web server interconnessi a livello mondiale dà vita al World Wide Web.

HTTP

In telecomunicazioni e informatica l'**HyperText Transfer Protocol (HTTP)** (protocollo di trasferimento di un ipertesto) è un protocollo a livello applicativo usato come principale

sistema per la trasmissione d'informazioni sul web ovvero in un'architettura tipica client-server.

Le specifiche del protocollo sono gestite dal World Wide Web Consortium (W3C).

Un server HTTP generalmente resta in ascolto delle richieste dei client sulla porta 80 usando il protocollo TCP a livello di trasporto.

TCP

In telecomunicazioni e informatica il **Transmission Control Protocol (TCP)** è un protocollo di rete a pacchetto di livello di trasporto, appartenente alla suite di protocolli Internet, che si occupa di controllo della trasmissione ovvero rendere affidabile la comunicazione dati in rete tra mittente e destinatario.

Su di esso si appoggia gran parte delle applicazioni della rete Internet.

Porte

Le **porte**, nella telematica e nell'ambito delle reti di calcolatori, sono lo strumento utilizzato per realizzare la moltiplicazione delle connessioni a livello di trasporto, ovvero per permettere ad un calcolatore di effettuare più connessioni contemporanee verso altri calcolatori, facendo in modo che i dati contenuti nei pacchetti in arrivo vengano indirizzati al processo che li sta aspettando.

In particolare il termine porta è mutuato direttamente da quello di porta fisica di connessione tra cavi di un calcolatore o di un qualunque altro dispositivo elettronico, ma, di fatto, nell'ambito della trasmissione dei dati durante una connessione non c'è altra porta fisica oltre a quella di aggancio del connettore del cavo.

Nell'ambito delle reti di telecomunicazioni il termine porta indica quindi solamente una porta virtuale o logica e può essere vista banalmente come un'etichetta (label) che identifica e discrimina il traffico dati di una connessione.

Tutti i processi di rete tra i vari terminali di rete necessitano dunque di una porta per l'instaurazione, l'esecuzione ed il completamento del servizio preposto.

Architettura multi-tier

Nell'ingegneria del software, il termine **architettura multi-tier** o **architettura multi-strato** (spesso definita con l'espressione inglese **n-tier architecture**) indica un'architettura software in cui le varie funzionalità del software sono logicamente separate ovvero suddivise su più strati o livelli software differenti in comunicazione tra loro (nel caso di applicazioni web questi strati sono la logica di presentazione, l'elaborazione dei processi e la gestione della persistenza dei dati).

CSS

Il **CSS** (acronimo di **Cascading Style Sheets**, in italiano **fogli di stile a cascata**), in informatica, è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML come, ad esempio, i siti web e relative pagine web.

Le regole per comporre il CSS sono contenute in un insieme di direttive (*Recommendations*) emanate a partire dal 1996 dal W3C.

L'introduzione del CSS si è resa necessaria per separare i contenuti delle pagine HTML dalla loro formattazione o layout e permettere una programmazione più chiara e facile da utilizzare, sia per gli autori delle pagine stesse sia per gli utenti, garantendo contemporaneamente anche il riutilizzo di codice ed una sua più facile manutenzione.

JSP

In informatica **JavaServer Pages**, di solito indicato con la sigla **JSP** (letto anche talvolta come **Java Scripting Preprocessor**), è una tecnologia di programmazione Web in Java per lo sviluppo della *logica di presentazione* di applicazioni Web, fornendo contenuti dinamici in formato HTML o XML.

Codice sorgente

Il **codice sorgente** (spesso detto semplicemente **sorgente** o **codice** o **listato**), in informatica, è il testo di un algoritmo di un programma scritto in un linguaggio di programmazione da parte di un programmatore in fase di programmazione, compreso all'interno di un file sorgente.

Esso definisce dunque il flusso di esecuzione del programma stesso.

Esso dovrà essere opportunamente elaborato per arrivare a un programma eseguibile dal processore ponendosi dunque come punto di partenza ("sorgente") dell'intero processo che porta all'esecuzione del programma stesso da parte dell'hardware della macchina, e che può includere altre fasi come precompilazione, compilazione, interpretazione, caricamento e linking (a seconda del tipo di linguaggio di programmazione utilizzato) per concludersi con l'installazione.

Per estensione, l'espressione "*codice sorgente*" viene utilizzata anche per riferirsi a testo scritto in linguaggi informatici non di programmazione, come i linguaggi di markup (HTML, XML e via dicendo) o i linguaggi di database (es. SQL).

PHP

PHP (acronimo ricorsivo di "PHP: Hypertext Preprocessor", preprocessore di ipertesti; originariamente acronimo di "Personal Home Page"^[1]) è un linguaggio di scripting interpretato, originariamente concepito per la programmazione di pagine web dinamiche. L'interprete PHP è un software libero distribuito sotto la PHP License.

Attualmente è principalmente utilizzato per sviluppare applicazioni web lato server, ma può essere usato anche per scrivere script a riga di comando o applicazioni stand-alone con interfaccia grafica.

Un esempio di software scritto in PHP è MediaWiki, su cui si basano i progetti wiki della Wikimedia Foundation come Wikipedia.

ASP

L'**application service provider** (ASP) è un modello architetturale per l'erogazione di servizi informatici. Il modello architetturale prevede che la tecnologia di elaborazione (hardware) e quella applicativa (software) vengano gestite centralmente presso un service provider lasciando all'utente finale la scelta dei tempi e dei modi di fruizione del servizio. Tipicamente, lo strumento software lato cliente che funge da interfaccia con il servizio applicativo è il web browser.

I vantaggi sostanziali di un tale tipo di servizio si ritrovano in un risparmio di costi da parte del cliente (es. manutenzione hardware e software on-site) che dovrà pagare l'utilizzo del servizio che include i costi di licenza dei programmi, di manutenzione dell'hardware ecc.

I dati elaborati dal cliente possono venir memorizzati sull'infrastruttura storage del fornitore oppure localmente.

Gli applicativi più diffusi con questo tipo di tecnologia sono gli ERP, i CRM, gli applicativi di e-commerce, di e-procurement, di data warehousing, di business intelligence e, in taluni casi, applicativi per l'automazione di ufficio.

Il termine ASP è stato rimpiazzato in tempi recenti da SaaS (Software as a service).

Web browser

In informatica il **web browser** (o più semplicemente **browser**, traducibile come *navigatore*) è un'applicazione per l'acquisizione, la presentazione e la navigazione di risorse sul web. Tali risorse (come pagine web, immagini o video) sono messe a disposizione sul World Wide Web (la rete globale che si appoggia su Internet), su una rete locale o sullo stesso computer dove il browser è in esecuzione.

Il programma implementa da un lato le funzionalità di client per il protocollo HTTP, che regola il download delle risorse dai server web a partire dal loro indirizzo URL; dall'altro quelle di visualizzazione dei contenuti ipertestuali (solitamente all'interno di documenti HTML) e di riproduzione di contenuti multimediali (rendering).

Tra i browser più utilizzati vi sono Google Chrome, Internet Explorer, Mozilla Firefox, Microsoft Edge (uscito con Windows 10), Safari, Opera e Maxthon.

CMS

In informatica un **content management system**, in acronimo **CMS** (*sistema di gestione dei contenuti* in italiano), è uno strumento software, installato su un server web, il cui compito è facilitare la gestione dei contenuti di siti web, svincolando il webmaster da conoscenze tecniche specifiche di programmazione Web.

Esistono CMS specializzati, cioè appositamente progettati per un tipo preciso di contenuti (una wiki, un blog, un forum, un giornale online, ecc.) e CMS generici, che tendono a essere più flessibili per consentire la pubblicazione di diversi tipi di contenuti.

Quando si parla di "contenuti" si intendono dati, informazioni, conoscenze, in qualsiasi forma (testo, immagine, video, musica, software, ecc.).

Quando si parla di "struttura" s'intende: impaginazione, formattazione, architettura, sezioni, gestione delle modifiche e delle versioni, gerarchia e privilegi dei ruoli. Tecnicamente, un

CMS è un'applicazione lato server che generalmente si appoggia su un database per l'archiviazione dei contenuti.

6. Sitografia

public.web.cern.ch.
w3.org.
it.wikipedia.org/wiki
webmasterpoint.org.
corsidia.com.
mrwebmaster.it.
validator.w3.org.
ecma-international.org.
research.nihonsoft.org.
netscape.com.
pear.php.net.
tcllib.sourceforge.net.
search.cpan.org.
jquery.com
github.com
jqueryitalia.org.
packages.debian.org
linuxjournal.com.
msdn.microsoft.com.