**TECNOLOGIE WEB**

Fabio Vitali

22/2

INTRODUZIONE AL CORSO

Argomenti del corso:

le basi: http, URI, Codifica dei caratteri

il web dei documenti: XML, WCAG, CSS, HTML, Markup

il web dei programmi: Server-side (PHP, Python, NodeJs), Client-side (Javascript, JS framework, Components, ARIA), WebService (REST)

il web dei dati: Linked Data Ontologie, SPARQL, JSON-LD, RDF

Esame:

- compito scritto con domande di teoria + esercizi per tutti (da fare) e domande 9 cfu (quindi da non fare) -> esame a libri aperti!!!! (70%)

Appelli: **metà giugno**, inizio luglio, metà luglio, settembre, gennaio 2022, febbraio 2022

*Pallini del prof: sapere HTML, con studio individuale (verrà richiesto più di quanto verrà spiegato)*

- progetto in gruppo di 2 o 3 persone + valutazione orale del contributo personale al progetto (35%)

Ogni team porta il progetto insieme, il team dichiara in anticipo la natura del contributo di ciascun membro oppure accetta che chiunque sia interrogato su tutto il progetto

Il riuso di codice, principi, approcci, architetture è importante, non vanno reinventati

L’interoperabilità è importante, i servizi messi a disposizione da altri possono essere utilizzati all’interno dell’applicazione 🡪 MASHUP per cui i servizi sono ancora più sofisticati di quelli dei componenti base

L’applicazione deve funzionare, essere utilizzabile

DICHIARATIVITA’ 🡪 l’algoritmo deriva in maniera quasi automatica dalla diretta descrizione degli input e output che voglio ottenere (la progettazione dichiarativa descrive stati iniziali e finali di un mondo astratto)

SEMANTICITA’ 🡪 un’applicazione semantica fa molto di più che visualizzare: collega, attiva, rende eseguibili le informazioni in maniera ricca e complessa

ACCESSIBILITA’ 🡪 permettere a qualunque persona indipendentemente dalle limitazioni fisiche o digitali di fruire l’applicazione (progettare per la disabilità significa anche progettare per le persone normodotate in situazioni particolari che li avvicinino ai disabili)

**NOLONOLOPlus** (sistema gestionale per un’azienda di noleggi)

Storage: modo in cui memorizzo i dati in modo che siano memorizzati anche una volta tornati sull’applicazione 🡪 include ANAGRAFICA CLIENTI, INVENTARIO DEI BENI con descrizioni, prezzi, stato ecc., CALENDARIO DEI NOLEGGI con storico e futuro, FATTURAZIONE DEI NOLEGGI con totali e statistiche

Presentation logic: come la cosa viene presentata a schermo

In mezzo, c’è la APPLICATION LOGIC: dove si gestisce quali operazioni sono permesse e come il dato viene ricevuto, generato e trasformato 🡪 in molti casi il rapporto tra dato memorizzato e presentato è molto difficile da comprendere 🡪 vero scopo del progetto

L’applicazione può garantire alcuni servizi aggiuntivi: suggerimenti di alternative, di estensioni o noli aggiuntivi, recensioni, gestione dei clienti regolari, proposta di noleggi ulteriori associati ecc.

EXPLAINABLE SOFTWARE: software la cui logica interna è basata su un modello che possa essere chiaramente espressa in un linguaggio umano

Spiegare significa

* comprendere l’output del software per debugging e miglioramento della performance dello strumento
* aiutare l’utente a comprendere il comportamento dello strumento e a costruire fiducia
* progettare strumenti interpretabili e inclusivi che aiutino a individuare e risolvere complessità, deformazioni, eccessi e altri difetti nei dati e nei modelli
* migliorare la trasparenza e la affidabilità degli strumenti

Non siamo un’azienda di noleggio, ma un’azienda che sviluppa software gestionale per aziende di noleggio.

Bisogna tenere conto della tipologia degli oggetti, della disponibilità nel tempo, dei fermi obbligati, dei prezzi che variano a seconda della stagione e della quantità di richieste ecc. 🡪 non lineare

L’applicazione ha 3 facce:

- il FRONT OFFICE (per il cliente dell’azienda di noleggio), mobile first, con noleggio, restituzione, prenotazione, fatturazione in maniera chiara, intuitiva e comprensibile

- il BACK OFFICE (per gli impiegati dell’azienda che non sono informatici), dove si inseriscono e rimuovono gli oggetti, gestiscono prezzi e descrizioni -> può essere solo per pc

- la MANAGEMENT DASHBOARD (per il manager) dove si generano overview e statistiche e si controlla l’andamento complessivo dell’azienda

Pattern: meccanismi standard per la risoluzione dei problemi # Anti-pattern: soluzione sbagliata ad un problema comune (es. cargo cult programming, coding by exception, copy and paste programming, hard coding, magic numbers, reinventing the wheel, giocare a poker con quattro carte)

24/02

STORIA DEL WWW

SISTEMA IPERTESTUALE -> sistema che rende espliciti i riferimenti tra un documento e l’altro, generati separatamente e indipendentemente gli uni dagli altri (es. libri della Divina Commedia – rapporto testo e commento 30/70)

Vannevar Bush, consulente scientifico della presidenza degli USA negli anni ’40, fa partire il progetto Manhattan. Era ingegnere e faceva sistemi elettromeccanici basati su microfilm (per essere letto aveva bisogno di lenti di ingrandimento molto potenti) -> meccanismo primario per ridurre dimensionalmente gli archivi di dati -> Ipotizza una macchina che potesse creare collegamenti tramite una punzonatura che permetteva il riconoscimento e quindi la ricerca rapida di microfilm (Memex, non fu mai realizzata)

Ted Nelson, visionario autore di “Computer Lib” che, dichiarando cose inaudite, fu uno dei libri più importanti per il successo dei personal computer. Bill Gates fonda Microsoft dopo aver letto il libro per lavorare sul personal computer. L’altra metà del libro si chiamava “Literary Machines”, illustrava un sistema integrato di gestione della letteratura ovvero dei contenuti, dei riferimenti espliciti ed impliciti, e del processo che sostiene la produzione letteraria -> Xanadu, anch’esso mai realizzato, fu d’ispirazione per Tim Berners-Lee per il WWW

Douglas Engelbart, ha inventato il mouse, il sistema a finestre, l’editing contemporaneo, i sistemi di teleconferenza.

Tim Berners-Lee e Robert Cailliau, nel 1989 insieme al resto del gruppo di ricercatori, vennero incaricati di realizzare un meccanismo per la diffusione rapida di articoli, appunti e opinioni tra i fisici che ruotavano intorno al centro. -> identificano Internet, SGML (Standard Generalised Markup Language, da cui deriva l’HTML) e sistema ipertestuale. Alla conferenza del 1991 sugli ipertesti mostrarono con poco successo il primo prototipo, realizzato in cliemt-server: gli ipertestualisti non lo apprezzano ma piace ai fisici-nucleari

Nell’ottobre 1992 la NCSA (national centre for supercomputing applications), che si occupa di scovare idee e progetti non del tutto funzionanti per migliorarli, accoglie il progetto e ne fa una versione propria: server NCSA e browser Mosaic -> rinominato Netscape, diventa Corporation nel 1993 con Marc Andreessen e Jim Clark (industriale californiano che cercava una nuova idea da finanziare, co-fondatore della Silicon Graphics) -> successo immediato e totale

Microsoft dopo una falsa partenza con Microsoft Network, abbraccia la tecnologia internet e realizza un browser (internet Explorer) ed un server (Microsoft Information Server), il browser è già installato sui prodotti Microsoft quindi spesso utilizzato per comodità -> PRIMA GUERRA DEI BROWSER. Tra Nescape e Explorer.

Nasce il World Wide Web Consortium (W3C), organizzazione priva di scopo di lucro. Come tavolo comune per non creare separatismi Netscape/Microsoft –> organismo di standardizzazione che per lungo tempo rappresenta l’autorità ufficiale di quello che è o non è web. HTML determina 16 colori. EXPLORER più di 200 colori.

La guerra dei browser continua fino a che Netscape ammette la disfatta nel ‘98, e rilascia il suo codice sorgente in OpenSource. Nasce una comunità online Mozilla.org (Communicator) e Netscape viene venduta a America On Line. Da una costola di Communicator nasce Firefox.

’97 -> versione definitiva di HTML -> XHTML

L’evoluzione del linguaggio per il W3C si occupa della pulizia e semplificazione della sintassi rigida e precisa basata su XML # a WHAT, gruppo che raccoglie sviluppatori di browser per il web, non interessa la semplificazione di linguaggi che sanno già gestire -> SECONDA GUERRA DEI BROWSER

Alla fine del 2008 nasce Chrome di Google che acquisisce rapidamente clientela e diventa il principale browser a partire dalla seconda metà del 2012

2015 Microsoft Edge, che non ha molto successo, nel 2020 esce una versione basata su Chrome

HTML Living Standard, in cui esiste un continuo flusso di sviluppo a tempi bisettimanali, i vendor decidono quali introduzioni di linguaggio fare e quando

Nelle guerre dei browser i contendenti cercavano di catturare gli utenti finali

Lock-in tecnologico: situazione in cui ci sono progettisti di applicazioni tradizionali (html css javascript), angular (religiosamente pure e precise), react (completamente diverso)

Realtà ben separate

INTRODUZIONE AL WWW

WWW -> Sistema integrato per la presentazione a schermo di documenti multimediali e per l’utilizzo di link ipertestuali per la navigazione

Sistema aperto a partecipazione volontaria distribuito e scalato su tutta Internet, basato su alcuni semplici concetti:

- il client o browser è un richiedente di informazioni (visualizzatore di documenti ipertestuali e multimediali), di fatto diventa un esecutore generico di applicazioni di qualunque tipo

- il server possiede o è in grado di confutare queste informazioni (diventa un interaccia tra i richiedenti e applicazioni server side che generano dei contenuti

3 protocolli fondamentali del WWW:

- uno standard per identificare in maniera generale risorse (che non sempre sono file) di rete e per poterle specificare all’interno di documenti ipertestuali (chiamato URI, sintassi stringa che identifica esattamente una risorsa)

- un protocollo di comunicazione che permette di creare dei dialoghi che fanno fare navigazione da una risorsa all’altra + caricare contenuti per creare cose più o meno ricche (chiamato HTTP)

- specifico linguaggio di markup attraverso il quale vado a caratterizzare i contenuti testuali (HTML/XHTML)

Marc Andreessen abbandonò l’editing -> Mosaic e poi Netscape erano semplici browser, si inventò che l’immagine potesse essere messa in linea con il testo (grafica come decorazione e non solo metodo informativo messo in una finestra separata) -> non si distingue più tra contenuto e stile

Nel ’94 Netscape introduce un primo linguaggio di Client Scripting: LIVE SCRIPT poi ribattezzato JAVASCRIPT per sfruttare la moda di Java, introdotto come puro meccanismo aggressivo nei confronti di Microsoft

Microsoft risponde con VBSCRIPT e JSCRIPT (quasi completamente compatibile con Netscape)

ECMA (fa standard di elettronica) standardizza il linguaggio.

L’uso di trucchi per forzare HTML a rese grafiche insolite da parte dei graphic designer ha portato a creare linguaggi appositi per gestire gli aspetti di visualizzazione del documento -> CSS permette di controllare le caratteristiche dei documenti HTML (separazione tra aspetto strutturale/contenutistico che rimane di competenza di HTML e quello tipografico/grafico di CSS)

CSS non è associato a HTML, funziona con qualsiasi linguaggio di markup, mi permette di fornire controllo sia all’autore sia al lettore della pagina grazie al meccanismo della cascata -> garantisce il supporto delle necessità particolari dei singoli utenti (es. ipovedenti) + controllo del designer sulla sua pagina

HTTP è state-less, ha una richiesta e una risposta, fine. Dialoghi più lunghi non possono essere realizzati -> Netscape crea i cookie (meccanismi per la gestione dello stato, salvataggio di ciò che è stato fatto finora)

In questo contesto il sito web è statico

Le applicazioni server-side CGI-BIN generano il concetto di sito web dinamico

-> LAMP (Linux Apache MySQL Perl), stack di linguaggi e protocolli per la realizzazione di siti web dinamici e interattivi, con memorizzazione delle informazioni su un database relazionale, separazione di memorizzazione, logica e distribuzione

-> REST (representational state transfer) stile di progettazione delle modalità con cui metto a disposizione dei servizi che possono essere interrrogati e ottenuti, usa correttamente la sintassi degli uri e la semantica dell’http

-> SEMANTIC WEB attribuzione di semantica ai dati per la creazione e la condivisione di un modello concettuale delle informazioni

-> AJAX (Asynchronous Javascript And XML, dove xml era il modo standard di generare applicazioni), meccanismo asincrono per generare applicazioni web client-side e server-side fortemente interattive e in grado di minimizzare il traffico di rete, trasforma il modello di applicazioni da LAMP del sito web dinamico, ad un meccanismo per cui caricando una pagina una volta per tutte lo script la aggiorna quando necessario

-> Node.js, l’applicazione ha non solo la parte server-side ma anche Javascript client-side. Insieme a npm, il loader di moduli per Node, ha creato una varietà ricchissima di estensioni e servizi per applicazioni web mai viste prima (ho a disposizione una libreria vastissima di funzionalità)

-> MEAN/MERN stack (MongoDB, ExpressJS Angular/react.js, NodeJs), nuovi stack di linguaggi e tecnologie per il web, di gran moda adesso

Database relazionale tradizionale -> ogni concettualizzazione di entità che mi interessa descrivere è rappresentata con una griglia (table) di proprietà (colonne) e identità (righe) + relazioni tra una tabella e l’altra

Mobile first ->

01/03

URI

03/03

CHARACTER ENCODING

08/03

HTTP

15/03