



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Introduzione al corso di Tecnologie Web - A.A. 2021/22

Fabio Vitali

Corsi di laurea in Informatica e
Informatica per il Management

Alma Mater – Università di Bologna

Introduzione

Oggi esaminiamo in breve:

- Informazioni pratiche sul corso
- I contenuti del corso
- L'organizzazione del progetto
- La tipologia d'esame



Docente delle lezioni

- Fabio Vitali
- Ricevimento:
 - Prima e dopo la lezione
 - Oppure per e-mail: fabio.vitali@unibo.it



Altre lezioni da parte di

- prof. Angelo Di Iorio
angelo.diiorio@unibo.it
- dott. Francesco Sovrano
francesco.sovrano2@unibo.it
- dott. Vincenzo Rubano
vincenzo.rubano@unibo.it



Quanti crediti? Quante ore? Quale progetto? Quale esame?

- Da quest'anno, Tecnologie Web è solo per la triennale di Informatica e vale 9 CFU.
- Esiste un corso di Tecnologie Web per Informatica per il Management da 6 CFU tenuto dal prof. Angelo Di Iorio.
- I due corsi vanno paralleli, e il prof. Di Iorio farà qualche lezione per voi mentre io andrò a fare qualche lezione per lui.
- Anche progetto ed esami proseguono in maniera parallela, fermo restando che Informatica per il Management avrà meno ore e meno contenuti di noi



Orario delle lezioni (1)

L'orario delle lezioni è:

- Lunedì, 15:00-18:00, Aula Tonelli (Matematica)
- Mercoledì, 16:00-18.00, Aula G1 (Geologia) - Esercitazioni
- Giovedì, 13:00-16:00, Aula G1 (Geologia)



Orario delle lezioni (2)

Questo è un corso di 72 ore, organizzate in 8 ore alla settimana, su un arco di 12 settimane (esclusa Pasqua).

N.B.: Come *forse* sapete, succede che

$$12 \times 8 = 96 \gg 72$$

Abbiamo dunque un po' di margine per saltare lezioni, gestire assenze e malattie, e finire un po' prima.

Ad esempio, nelle prime due settimane non faremo esercitazione (lezione del mercoledì)



Organizzazione delle lezioni

- Lezioni di teoria (in aula) LUN e GIO
- Lezioni di esercitazione
(in aula con vostri portatili) MER
 - L'aula NON HA ABBASTANZA PRESE ELETTRICHE PER TUTTI.
 - Portatevi delle ciabatte elettriche



Sito Web del corso

Il sito

<https://virtuale.unibo.it/course/view.php?id=26726>

conterrà:

- Lucidi PowerPoint e PDF
- Annunci vari
- Link a documenti disponibili su rete

Registrerò le lezioni (***ma ricordatemele***) e le metterò a disposizione su OneDrive.



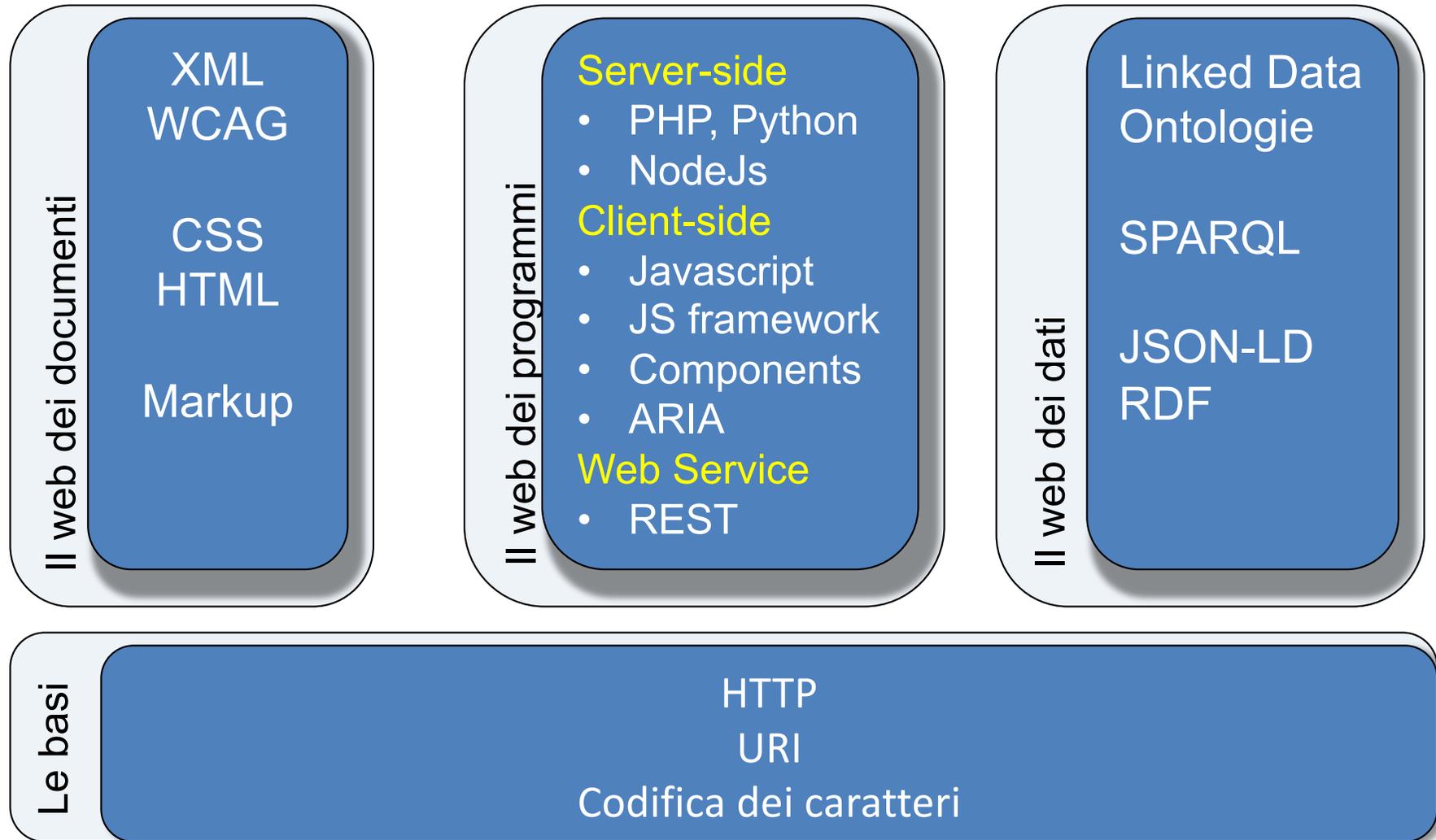
Le lezioni

Organizzazione delle lezioni

- Su lucidi immediatamente a disposizione
- Autonome e complete
(ogni lezione un argomento diverso - circa)
- Enfasi sul significato delle tecnologie, piuttosto che sugli strumenti
- Indicazione della bibliografia
 - utilizzata per i lucidi (testo normale)
 - suggerita per chi volesse approfondire (testo corsivo)
 - richiesta per l'esame (testo corsivo e grassetto)



Argomenti delle lezioni



Com'è fatto l'esame

- E' necessario partecipare ad un progetto per team durante l'anno.
- L'esame consiste in:
 - Valutazione orale del contributo personale al progetto
 - Sfrutta le tecnologie spiegate nel corso
 - Evidenzia i problemi di mashup tra tecnologie diverse
 - Compito scritto di 5/6 domande.
 - Orale in casi eccezionali



Valutazione delle prove

Scritti e progetti vengono valutati in trentesimi.

- Lo scritto pesa l'70% del voto finale
- Il progetto base pesa il 35% del voto finale
- L'aggiunta di funzionalità facoltative porta il peso fino al 40% del voto finale

Sia m il voto dello scritto e n il voto del progetto, sia p il peso del progetto ($p \in [0.35, 0.40]$), il voto finale sarà dato da:

$$0.7*m + p*n$$

Ad esempio, con 26 allo scritto e 28 al progetto base:

$$0.7*26 + 0.35*28 = 18.2 + 9.8 = 28$$

Ad esempio, con 27 allo scritto e 28 al progetto completo:

$$0.7*26 + 0.40*28 = 18.9 + 11.2 = 30.1$$



Appelli d'esame

- Uno a inizio giugno per chi è in debito dall'anno scorso.
- Uno a metà giugno (*in cui mi aspetto MOLTI di voi*)
- Uno a inizio luglio
- Uno a metà luglio (*nella settimana delle tesi*)
- Uno a settembre
- Uno a gennaio 2022
- Uno a febbraio 2022



Organizzazione dei team

- Ogni persona decide in anticipo se è interessata a sostenere l'esame in estate, autunno, sessione straordinaria o indeciso.
- Tutti gli studenti si dividono in team di 2-3 persone. Non sono accettati gruppi di più di 3 persone. Non sono accettati progetti singoli tranne eccezioni estreme e ben giustificate.
- Ogni team porta il progetto insieme (non ci sono eccezioni!). Il team dichiara in anticipo la natura del contributo di ciascun membro oppure accetta che chiunque sia interrogato (e nel dettaglio!) su tutto il progetto.
- Io non sono coinvolto nell'organizzazione dei team.



Pallini del docente

In generale

- HTML bisogna saperlo e basta
(Nota: non verrà spiegato nel dettaglio che verrà richiesto)
- C'è enfasi sul web come produttore di contenuti e sull'approccio dichiarativo.
- Il content model misto è utile
- Giusto compromesso tra competenza pratica e studio teorico
- Il progetto **non** fornisce tutte le competenze per passare l'esame.

Nei progetti

- Il riuso (di codice, principi, approcci, architetture) è importante
- Un buon stile di programmazione è importante
- La capacità di rendersi indipendente dagli strumenti è importante
- L'interoperabilità è importante
- I portapizze non sono tollerati
- Gli smanettoni sono premiati solo se studiano



Le parole chiave del corso

- Interoperabilità
- Standard
- Mashup
- Dichiaratività
- Semanticità
- Accessibilità



Standard e interoperabilità

- La prima informatica era contenta di un un'applicazione che funzionasse e basta.
- Con il tempo si sono introdotti concetti di qualità dell'implementazione: efficienza nell'uso delle risorse, complessità computazionale, gestibilità nel tempo del codice sorgente, usabilità, utilizzo appropriato delle librerie, ecc.
- A noi interessa l'interoperabilità, un nuovo aspetto della progettazione: non solo che il programma funzioni e che funzioni bene (per interpretazioni specifiche di "bene"), ma che funzioni con altri programmi dello stesso tipo.
- La identificazione dei standard appropriati, la corretta implementazione degli stessi, l'identificazione di sempre nuove e più precise aree di interoperabilità sono concetti fondamentali per questo corso



Mashup

- La realizzazione di siti, applicazioni, servizi Web richiede la padronanza di un alto numero di concetti, linguaggi, protocolli, ciascuno dei quali è spesso semplice, espressivo, potente.
- La difficoltà in questo caso è trovare il modo per combinare e congiungere questi concetti ecc. in una costruzione ampia e complessa, che fornisca servizi ancora più sofisticati di quelli dei componenti base.
- Questo è quello che intendiamo per mashup



Dichiaratività

- La progettazione imperativa o procedurale prevede la esplicita somministrazione di istruzioni ad una macchina ubbidiente e cieca.
 - Questo porta con sé inevitabilmente rischi e problemi (sintattici, semantici, ecc.)
 - Un errore procedurale interrompe l'esecuzione.
- La progettazione dichiarativa descrive stati iniziali e finali di un mondo astratto
 - Un meccanismo esistente genera la computazione che permette la transizione dallo stato iniziale al finale
 - Un errore dichiarativo porta ad uno stato finale non desiderato, non ad un'interruzione.



Semanticità

- Il web non è più (non è mai stato) il mondo della visualizzazione pura e semplice di informazioni testuali o, peggio, strutturate.
- Le informazioni significano, e attivano ruoli, funzioni, processi rilevanti e dipendenti dalla loro natura e dal loro contenuto.
- Un'applicazione semantica fa molto di più che visualizzare: collega, attiva, rende eseguibili le informazioni in maniera ricca e complessa.



Accessibilità

- Il web rappresenta un mezzo di comunicazione universale, a basso costo, in grado di superare ogni distanza, ogni classe sociale, ogni differenza di ceto e età.
- Con pochi sforzi aggiuntivi, i progettisti di siti ed applicazioni web possono assicurarsi che anche i disabili possano utilizzare a pieno il web e i suoi servizi.
- Progettare per la disabilità significa anche progettare per le persone normodotate in situazioni particolari che li avvicinino ai disabili: ad esempio, usare il web con gli occhi occupati alla guida, o in un ambiente rumoroso, o in una situazione dove non è possibile usare le mani.



Il progetto

- Un sistema VERO, che funziona e fa cose utili
- Realizzabile sia in laboratorio che a casa.
- Enfasi in parte sulla programmazione (approccio procedurale) ma soprattutto sui documenti attivi (approccio dichiarativo)
- Enfasi sul mashup di tecnologie esistenti e sofisticate



Animal House

Un sistema gestionale per una catena di negozi e pensioni VIP per animali domestici.



*I nostri ospiti
a quattro zampe*



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



Background: strumenti per lo sviluppatore web negli anni '20

Lo sviluppo sw più sorprendente ed evidente degli ultimi anni è la nascita di innumerevoli strumenti per il programmatore, con cui realizzare servizi sofisticati

- ❖ Framework: invece di scegliere nuovi linguaggi di programmazione, si usano ora librerie per gli scopi più disparati, facilmente integrabili e mescolabili tra loro
- ❖ API (Application Programming Interfaces): invece di sviluppare applicazioni monolitiche che svolgono servizi complessi in un'unica maniera, si forniscono meccanismi di manipolazione delle strutture dati fondamentali e accesso agli algoritmi più sofisticati per applicazioni sviluppate dai clienti.

Questo permette incredibile sofisticazione, grande componibilità, e rapidità di sviluppo precedentemente irraggiungibili.





I framework

- I framework sono librerie che rendono più ricco, sofisticato e semplice l'uso di una tecnologia, come un linguaggio server-side, un linguaggio client-side o le specifiche grafiche di una pagina web.
- Server-side esistono dalla fine degli anni novanta, e hanno reso la programmazione a tre livelli drasticamente più facile.
- Client-side si sono sviluppate a partire dal 2002, su CSS e Javascript, con scopi molto difforni.





Application Programming Interface (API)

- Librerie, protocolli e strumenti per permettere di utilizzare gli algoritmi ed i servizi messi a disposizione da un software da parte di un altro software, invece che da esseri umani.
- Applicazioni e servizi che forniscono una API delegano ad un'applicazione terza aspetti come interazione, interfaccia, navigazione, ecc. e forniscono via API il solo servizio nudo.
- Uno degli scopi tipici di ricorso ad API è per integrare più servizi in un'applicazione più ricca e potente di quelle utilizzate come base.
- Questo si chiama *mashup* ed è una delle caratteristiche più evidenti di questo periodo storico: mescolare servizi di base per ottenere applicazioni imprevedute dai fornitori dei servizi stessi.





Le API REST

- Le API che ci interessano sono più concretamente quelle che vanno sotto il nome di RESTful API, ovvero che sfruttano al meglio la natura di HTTP e degli URI (i protocolli più importanti del web) per fornire i loro servizi.
- Una API RESTful fornisce:
 - Un URI base a cui accedere per ottenere i servizi
 - Una sintassi degli URI delle entità interrogabili e modificabili
 - Un media type attraverso cui ottenere e fornire dati da utilizzare nei servizi forniti (ad esempio XML, JSON, etc.)
 - Una semantica associata all'uso dei vari verbi HTTP (GET, PUT, POST, DELETE) attraverso i quali attivare e eseguire i servizi offerti.
- La documentazione di un'API tipicamente descrive nel dettaglio nomi (URI), verbi (comandi HTTP) e formati (Internet Media Type) per ogni servizio fornito.





Le API di servizi locali

- Le seconde API che ci interessano sono quelle che permettono alle applicazioni web di accedere a speciali servizi offerti dal device su cui vengono eseguite (tipicamente accesso a periferiche di I/O particolari).
- La diffusione di TCP/IP su device mobili ha ampliato radicalmente la quantità e qualità di periferiche "non tradizionali" a cui l'applicazione può avere accesso:
 - Telecamera (Camera API)
 - Microfono e sintesi vocale (Speech API)
 - Geolocalizzazione (via GPS o altro) (Geolocation API)
 - Suoni (Web Audio API)
 - Vibrazioni (Vibration API)
 - Telefono (Web telephony API),
 - ecc. ecc.
- Ci sono circa 80 API diverse al momento utilizzabili sulla maggior parte dei browser (purché il device offra la funzionalità)





Animal House: background

- Fate parte di un'azienda che sviluppa software gestionale per prodotti e servizi per animali domestici. Il vostro cliente più importante è l'azienda Animal House, leader di settore.
- Animal House è una realtà importante e diffusa sul territorio: ha più di 80 sedi sparse per il territorio italiano, e offre una varietà di servizi in presenza e in remoto.
- I servizi offerti sono molteplici: prodotti per animali domestici, veterinari e psicologi per animali, pensioni per cani, vendita di animali di razza e mercato libero per i cuccioli dei propri clienti.
- Animal House offre anche servizi di fascia alta destinati a clienti danarosi e viziati: pensioni a 4 stelle, psicologi per animali, addestratori, wedding planner, ecc.
- Queste complessità possono rendere complesso e non banale rappresentare agli utenti la quantità e la qualità dei servizi per la clientela normale e quella "VIP".





Animal House: Presentation logic

- Dovete costruire un'applicazione per la gestione dei servizi da parte di Animal House.
- L'applicazione ha tre facce:
 - Il *game*, ovvero l'applicazione di intrattenimento che permette ai proprietari di animali domestici di avere informazioni e curiosità sui propri cari, fare giochi e confrontare prodotti dello store. Mobile first, solo online, nessun database.
 - Il *front-office*, ovvero l'applicazione per i clienti, mobile first, per permettere le attività di accesso e uso di tutti i servizi online e per la prenotazione di quelli in presenza per i clienti *regular*.
 - Il *back-office*, ovvero l'applicazione per gli impiegati dell'azienda che, sede per sede, inseriscono e rimuovono servizi e prodotti, gestiscono prezzi e descrizioni, ecc.).





Animal House: application logic

- Oltre alle funzionalità di semplice manipolazione e presentazione dei dati, l'applicazione dovrà/potrà garantire anche servizi aggiuntivi, come:
 - Suggerimenti di alternative
 - Suggerimenti di estensioni e prodotti aggiuntivi
 - Recensioni
 - Gestione di richieste ripetute da clienti regolari
 - etc. ...
 - Il front office deve avere due interfacce, clienti regular e VIP. L'interfaccia VIP dovrà essere particolarmente accattivante e piacevole, e il cliente VIP dovrà sentirsi coccolato e rispettato al 100%.
 - E' indispensabile che l'utente comprenda senso e funzionamento di questi servizi aggiuntivi e si senta invogliato a farne uso
- (*): L'elenco completo di funzionalità aggiuntive obbligatorie e facoltative verrà comunicato nelle specifiche definitive.





Servizi speciali per Animal House

- Intrattenimento
 - *Quiz* : esistono diverse fonti per caricare informazioni, ad esempio <https://zoo-animal-api.herokuapp.com/animals/rand/> o altre specializzate. L'applicazione genera dinamicamente le domande a partire da questi dati e fa giocare l'utente. Il backend memorizza i punteggi, lo storico o altro.
 - *Forse non sapevi che...* : suggerimenti di immagini e video sul tema (API YouTube o altre). Oppure notizie, letture, curiosità o altri contenuti reperiti su Web.
 - *Eccolo qua*: bacheca di condivisione immagini, aneddoti e altro.
- Servizi di "utilità"
 - *Di che razza è?* Riconoscimento di immagini attraverso servizio AI.
 - *Cerco compagnia*, sistema di accoppiamento controllato per l'animale di razza
 - *Cuccioli per tutti!* Mercatino cuccioli, gratis e a pagamento
 - *Whatsapp*: parla e guarda il tuo amico a quattro zampe anche quando è qui da noi





Il lavoro di team

- Tutti i membri dei team sono tenuti a lavorare e lavorare insieme.
- E' meglio essere parte attiva di un progetto mediocre che passiva di un progetto meraviglioso.
- Non saranno tollerati i portatori di pizze
- Mi riservo all'esame di scoprire il contributo individuale di ciascuno, indipendentemente dalla bontà del progetto consegnato.



Attenzione agli anti-pattern

- Durante la fase di analisi del progetto, sarà mia cura verificare che non siano state fatte sciocchezze implementative, in particolare anti-pattern (<http://en.wikipedia.org/wiki/Anti-pattern>).
- Alcuni anti-pattern da evitare assolutamente:
 - Cargo Cult programming
 - Coding by exception
 - Copy and Paste programming
 - Hard Code
 - Magic Numbers and Strings
 - Reinventing the wheel



Cargo-cult programming

Utilizzare un frammento di codice preso da un esempio o da una applicazione simile al proprio programma, senza prima aver capito cosa il codice faccia e come funzioni o, addirittura, senza avere neanche idea se possa servire a qualcosa o meno.



Coding by exception

Abusare del sistema di gestione delle eccezioni (casi isolati e rari per definizione) per lo svolgimento di compiti frequenti o normali o comunque appartenenti al flusso naturale dell'algoritmo.



Copy & Paste Programming

Produrre codice estremamente ripetitivo ottenuto dalla copia ripetuta dello stesso frammento senza alcuna variazione o con minime variazioni al suo interno, invece che attraverso un'appropriata strutturazione con funzioni chiamate ripetutamente.



Hard coding

Detto anche cablatura di valori o valori cablati

Introdurre in un codice sorgente alcuni valori costanti che non possono essere cambiati senza ricompilazione del codice sorgente e quindi irrigidiscono il programma a funzionare solo con quei codici



Magic numbers

Utilizzare un numero o una stringa di testo a cui non è associato un significato esplicito e riconoscibile, ma il cui valore è essenziale ai fini del funzionamento del codice.

Ad esempio utilizzare numeri invece di etichette riconoscibili in un codice oppure inserire URL assoluti nel codice



Reinventing the wheel

Ignorare una soluzione tecnica generalmente accettata e utilizzare al suo posto una soluzione ricreata localmente.

In un processo si progetta/sviluppa da zero elementi dell'architettura o di librerie esterne che già esistono e sono già ben testati e funzionanti.



Un anti-pattern in più

Non esiste questo pattern tra quelli noti in letteratura, ma dovrebbe esistere. Io lo chiamo

Giocare a poker con quattro carte

- Quattro carte bastano per fare poker. Quindi in teoria io posso lasciare una delle cinque carte sul tavolo e provare a fare poker solo con le prime quattro.
- Questo pattern lo si vede tra gli studenti quando provano a realizzare il progetto con forza bruta senza immaginarsi che esiste un linguaggio, un protocollo, un API che copre esattamente quel ruolo senza bisogno di reinventarselo.



Quando dare l'esame

- Ognuno deve decidere al più presto in quale sessione darlo e seriamente impegnarsi per mantenere questa decisione.
- Idealmente:
 - Estate 75-100%,
 - Autunno 20-0%,
 - Inverno 5-0%,
 - Indecisi 0 %.
- Quattro appelli a giugno/luglio, uno a settembre, due a gennaio/febbraio.
- Chi presenta il progetto entro febbraio 2022, anche se non riesce a dare lo scritto, non deve rifare il progetto ma solo studiare il nuovo programma.
- Chi non riesce a portare il progetto per febbraio 2022 deve prepararsi con il nuovo programma e realizzare il nuovo progetto



Suggerimenti per l'esame

- Venite alla presentazione con il progetto che funziona. Se non va io vi faccio tornare. Per questo preferisco
 - vedervi una settimana dopo l'appello con il progetto che funziona
- piuttosto che
 - perdere tempo con la presentazione di un progetto che non va,
 - mandarvi via in lacrime, e
 - vedervi una settimana dopo l'appello con il progetto che funziona
- Venite allo scritto avendolo preparato. Non c'è niente di più irritante di vedere ragazzi svegli e competenti (vi si riconosce) che prendono 8 o 10 allo scritto perché ci hanno solo provato.
- Lo scritto non è difficile per chi ha studiato, è impossibile per chi non l'ha fatto.



Attenzione all'appello di febbraio

- Quanto detto NON si applica all'appello di febbraio.
- Il 28 febbraio 2023 si conclude la possibilità di presentare il progetto di quest'anno.
- Non riducetevi all'ultimo, cercate di portare il progetto negli appelli estivi ed autunnali
- Gli slot a disposizione per presentare il progetto a febbraio non sono infiniti. Tutti gli anni questi slot (più del doppio degli altri appelli) si esauriscono molto presto. Garantisco slot di presentazione solo se fate richiesta entro il 31/12/2022.
- Io cerco di **non** essere più esigente a febbraio, ma inevitabilmente potrò dedicare meno attenzione al vostro meraviglioso progetto.
- Non riducetevi all'ultimo (l'ho già detto, lo so)



Flessibilità del corso

- Prova scritta e prova di progetto sono indipendenti.
 - Il progetto è sempre di gruppo
 - Lo scritto è sempre individuale
- Potete provare lo scritto tutte le volte che volete
 - Il voto precedente verrà cancellato solo se consegnate un nuovo scritto
 - Gli scritti sono solo alle date degli appelli ufficiali
- Potete presentare il progetto tutte le volte che volete
 - Solo se lo decide consensualmente TUTTO IL GRUPPO
 - Potete ritirarvi dalla presentazione del progetto in qualunque momento e tornare una settimana o due dopo con le correzioni che ritenete opportune.



Rigidità del corso

(nessuna eccezione per nessun motivo)

- **Il progetto deve funzionare.** Completamente ed esattamente secondo specifiche. Nel corso del tempo queste specifiche possono anche evolvere.
- **Il progetto deve risiedere su una macchina del dipartimento.** Questo include SIA il codice SIA tutti i dati del progetto (tranne quelli per cui è **esplicitamente** previsto l'uso di un server condiviso)
- **Potete installare librerie e SW per il progetto** a vostro piacimento, MA verificate prima che sia eseguibile sulle macchine e sui sistemi operativi offerti dal dipartimento e con i permessi d'uso di un utente normale,
- **Il progetto deve venire presentato da tutto il gruppo insieme**, in presenza (speriamo) oppure online su MS Teams. In nessun caso è accettabile che si presenti in una data una parte del gruppo e in una data diversa il resto del gruppo.
- ***Non usate tecnologie che richiedano compilazioni o procedure complesse di aggiornamento del codice eseguito.*** Deve essere possibile aprire un file sul server e modificarlo al volo durante la prova.
- ***Web packer, minimizzatori ed altre diavolerie simili NON sono gradite.*** Io preferisco vedere funzionare il codice che avete scritto e non una sua compilazione.



Conclusioni

- Il corso di TW vuole coprire alcuni aspetti fondamentali delle tecnologie e degli standard connessi con il World Wide Web.
- Il progetto vuole dare un'idea del lavoro d'équipe e del funzionamento dei team di standard, oltre a far provare con mano l'applicazione degli standard stessi.





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fabio Vitali

Dipartimento di Informatica – Scienze e Ingegneria
Alma mater – Università di Bologna

Fabio.vitali@unibo.it

www.unibo.it