* **Quali sono gli elementi base per la comunicazione web?**
  + URI: meccanismo per individuare e accedere alle risorse/documenti. È formato da URL (tramite indirizzo) e URN (tramite codice univoco)
  + Encoding: È la codifica del file trasmesso
  + protocollo http
  + richieste http
* **Com’é formato un URL?**

È formato da: //<hostname>:<porta>/<percorso-risorsa>?<query>#<frammento>

* **Cosa è un protocollo HTTP?**

È un’insieme di regole, che definiscono le modalità di trasferimento per un ipertesto. È usato come sistema principale per la trasmissione d’informazioni sul web, in un’architettura tipica client-server. La sua porta di default è la 80 e usa un meccanismo di messaggi di tipo richiesta e risposta. Si colloca nello strato Applicativo sopra il TCP/IP. È senza stato (STATELESS) senza relazione tra le richieste precedenti.

* **Chi gestisce le specifiche del protocollo HTTP?**

Sono gestite dal W3C (World Wide Web Consortium), la principale organizzazione internazionale standardizzata per il World Wide Web.

* **Cosa è un ipertesto?**

È un insieme di documenti, collegati tra loro tramite riferimenti ipertestuali, denominati link.

* **A cosa servono gli status code?**

Sono dei codici che vengono inviati dal server al client con la verifica della risorsa richiesta. Per esempio 200 se la risorsa richiesta era presente ed è stata ritornata senza problemi

* **Che differenza c’é tra HTTP e HTTP/2?**

HTTP/2 è la nuova versione del protocollo di rete HTTP basato su SPDY, che usa header compressi e tecnologie server push, migliorando il tempo di caricamento delle pagine.

**Architettura Web W3C**

Tecnologie Fondamentale Web

* Web data formats: HTML, …
* Web naming: URIs
* Web protocols: HTTP, …

Componenti Fondamentali Web

* Rappresentazione di stato di risorse (Status Code)
* Identificazione di risorse (URI)
* Protocolli che supportano l’interazione tra agenti e risorse nello spazio (HTTP(S), ecc)

**Linguaggi Web**

Quali sono i tipi di pagine web ?

* **Statiche:** Documenti residenti sul server Web il cui contenuto é definito al momento della loro creazione e non cambia più. Diverse richieste alla pagina danno la stessa risposta. (Files HTML)
* **Pagine Dinamiche:** Documenti Web creati dal Web Server solo quando uno user-agent (browser) li richiede. Il server attiva un programma/applicazione che crea dinamicamente il documento che viene ritornato. Contenuto può variare di richiesta in richiesta(pagine .php, asp, jsp …)
* **Pagine Attive:** Inter-aggive. Documenti che hanno un contenuto ed un’apparenza non completamente specificati dal server. Pezzi di programma vengono inclusi nel codice della pagina inviata dal Server al browser, e vengono eseguiti localmente sul client. Un documento attivo non é mai fisso, visto che può interagire con l’utente.

**Quali sono le tecnologie usate per creare pagine Web?**

* (X) HTML → Struttura pagina
* CSS → Layout
* JS → Interazione

**Che cos’è HTML?**

Linguagio di Marcatura ipertestuale. Etichette (<tag>) marcano elementi strutturali e di formattazione.

**Che cos’è XHTML?**

Extensible HyperText Markup Language. Famiglia di tipi di documenti che riproducono, come sottoinsieme o estensione HTML. Un file XHTML é una pagina HTML scritta in conformità con lo standard XML.

**Quali sono i vantaggi di XHTML rispetto a HTML?**

* Codice pulito e ben strutturato
* Portabilità
* Accessibilità
* Estensibilità
* **Quali sono le architetture di sistemi web?**

Client-server (2 Tiers), a 3 livelli (3 Tier) e a N-livelli (n Tiers). Sono dette anche Multi-Tier!

* **Che differenza c’é tra un architettura Client-server a quella 3 livelli (o N)?**

La separazione logica è suddivisa in presentazione (gui), logica e dati necessari per l’applicazione. A differenza del client-server che logica e presentazione sono insieme, aumentando il carico di processamento sul client. Con 3 livelli si ha una maggiore flessibilità nella progettazione e sullo sviluppo.

A N livelli si hanno comunque la suddivisione di UI, Logica e dati, ma con la differenza che ogni strato può essere maggiormente suddiviso (più DBMS, moduli in logica e diverse UI a dipendenza del client), dove sono presenti più computer in ogni livello.

* **Cosa è un Middleware?**

Software che consente ai componenti di un sistema distribuito (separazione a livelli) di interfacciarsi ed interagire. Ha lo scopo di comunicare tra i componenti, garantire il disaccoppiamento, permettere uno sviluppo indipendente dei un componente dagli altri e semplifica la riconfigurazione del sistema.

* **Che tipi di client esistono?**

Sono di tipo Thin (leggero) dove il client si appoggia ad un processamento remote centrale sul server. Ha bisogno di scarse risorse (SW e HW)

Thick (pesante) dove il processamento viene eseguito dai client e non si appoggia al server remoto, con una difficoltà maggiore per gli aggiornamenti futuri.

Smart (ricco) basato su modelli web-based invece di client/server, dove sono nodi connessi a internet che consentono all’utente di interagire con le applicazioni server mediante web services.

* **Cosa sono i Smart Client?**

Sono applicazioni facilmente rilasciabili e gestibili che offrono un esperienza interattiva ricca, adattiva e responsive sia in scenari online sia offline, sfruttando le risorse locali.

* **Cosa è una Progressive Web App?**

È l’unione del meglio del web e delle app, dove non richiedono un installazione, ma forniscono ai device mobile di avere un icona sulla home screen, ottenendo un’esperienza full screen con eventuali notifiche.

* **Quali sono I principi chiave di una Progressive Web App?**

Progressive, responsive, connettività indipendente dalle condizioni rete, App-like (UX come in un app), aggiornate, sicure, notificabili, installabili (home screen) e facilmente condivisibili.

* **Quali sono gli elementi fodamentali per il web?**

**Cosa è un web service?**

Interfaccia che descrive un insieme di operazioni accessibili in rete mediante messaggi standard in XML.

**Componenti di un Web Service?**

Descritto, pubblicato/pubblicizzato, scoperto, invocato e composto.

**Quali vantaggi ha un web Service?**

Interoperabilità, ubiquità, facilità e support di industrie a SOAP ed altre tecnologie.

**Standard per Web Service?**

Il fornitore costruisce e definisce il servizio usando WSDL, registra il servizio mediante Univeral Description Discovery and Integration (UDDI)

**Come funziona un server web?**

È un meccanismo di richiesta e risposta di una risorsa attraverso URI o XML type.   
(esempio Apache Tomcat)

**Come funziona un application server?**

È una tipologia di server che fornisce l’infrastuttura e le funzionalità di supporto, sviluppo ed esecuzione di applicazioni. Fornisce una serie di servizi come: template, sicurezza, persistenza dei dati, sessioni,….   
(es. TomEE Apache)

**Cosa è una servlet?**

Classi Java che gestiscono richieste e risposte HTTP, che operano all’interno di un server web o un application server. Alla richiesta della risorsa il container istanzia l’oggetto servlet e chiama il metodo specifico (GET, POST, HEAD…)

**Cosa è una Java Server Pages (JSP)?**

Tecnologia di programmazione web in Java per lo sviluppo della logica di presentazione di applicazioni Web, fondendo contenuti dinamici in formato HTML.

**Quali sono I component di Tomcat?**

* + Catalina: è il contenitore di servlet Java, che implementa le JavaServer Pages (JSP)
  + Coyote: è il connettore HTTP che ascolta le connessioni in entrata su una posta specifica TCP e processa le richieste.
  + Jasper: È il motore JSP che analizza I file JSP per compilarli in Java come servlet.

**Cosa è un Servlet Container?**

Componente di un web server che offre la comunicazione di support, la gestione del lifecycle, multithreading, security e support JSP.

**Cosa sono le JSP?**  
sono una tecnologia di programmazione in Java per lo sviluppo di logica di presentazione. È costituita da codice HTML con inserite sezioni di codice Java.

**Cosa è REST (Representational State Transfer)?**

È uno stile di architettura, non standard di sviluppo e non c’é nessuna specifica. Può essere usato come stile per progettare web service e prescrive l’uso di standards (HTTP, URL, XML…). Lo stile d’interazione pull-based Client-server (nodi client richiedono le rappresentazioni). Sono stateless, ossia che ogni richiesta deve contenere tutte le informazioni necessarie e sono nominate con URI. Le risorse sono interconesse mediante URI che permettono il passaggio da uno stato ad un altro.

**Quali sono I sei vincoli di REST?**

Interfaccia uniforme, stateless, cacheable, Client-server, sistema stratificato e Codifica on demand (op

**Quali sono I principali metodi HTTP?**

GET, POST, HEAD, PUT, DELETE, ….

**REST e HTTP access pattern (CRUD via HTTP)**

Create -> POST  
 Read -> GET  
 Update -> PUT  
 Delete -> DELETE

**Quali sono I passi di progettazione REST?**

Identificare le risorse, modellare le relazioni, definire URIs, comprendere il significato (get, post…), progettare e documentare, implementare e fare deploy su web server e testare via web browser.

Quali sono I codici risposta?  
 2xx -> Codici di successo  
 3xx -> Codici di redirezione  
 4xx -> Codice di errori  
 5xx -> Codice di errori server

**Cosa è SOA (service Oriented Architecture)?**

È un modo di progettare software che fornisce servizi ad utenti o ad altri servizi attraverso interface pubblice e ritrovabili. È un elemento base per il servizio web. È un design pattern ed ogni servizio e ben definite standard.

**Cosa è WOA (Web Oriented Architecture)?**

È un’area di SOA che focalizza l’uso di servizi su internet. Usa HTTP/HTTPS come protocollo per trasporto e REST o SOAP per l’invocazione e scambio di messaggi.

**Cosa è SOAP?**  
è un protocollo XML per scambiare informazioni, strutturata e tipizzata per il web. È basato su messaggi scritti in XML, modulabile ed estendibile, senza semantiche di applicazione ne di trasporto ed ha una sperazione del contenuto dal trasporto.

**Cosa definisce SOAP?**  
L’incapsulazione di dati da scambiare fra nodi (busta),

**Cosa è il web?**

Un insieme di nodi interconessi tra di loro

**Differenza tra REST e SOAP?**

REST: ottime API come leggibilità, difficle da estendere, non trasportano dati complessi di strutture fortemente tipizzati.

SOAP: Ottimo integrazione system-to-system, usano strutture dati precise e ben definite, documentazione API è piu semplice.

**Cosa è WSDL (Web Service Description Language)?**

È un vocabolario (documento) XML per descrivere la forma delle interface per la definizione delle web service. È uno standard W3C, ed è un document che descrive una WS in due parti: astratta e concreta.

La parte astratta descrive: I messaggi, le operazioni e un’interfaccia che raggruppa le operazioni senza vincoli di trasporto.

La parte concreta descrive: Un binding, un endpoint e un service che raggruppa endopoints che implementano la stessa interfaccia.

**Spring**

**Che cos’è un framework web?**

É un framework software progettato per supportare lo sviluppo di siti web dinamici, applicazioni web e servizi web.

**Qual’è lo scopo di un framework ?**

Risparmiare allo svilluppatore la riscrittura di codice già scrito in precedenza per compiti simili. Inoltre impongono al programmatore una precisa metodologia di sviluppo del software.

**Come funziona Spring MVC ?**

Spring MVC ha un Central Dispatcher risponsabile per rispondere a tutti i percorsi possibili dell’applicazione.  
La mappatura (associazione dell’indirizzo per una determinata risorsa) é fatta all’interno della classe tramite annotazione Java.  
(in .Net MVC inverce é fatta per convenzione, /sample/hello → chiamata ad una classe SampleController e al metodo hello)

Template Engine é uno strumento per generare file basiti su modelli. Spring può usare Thymeleaf.

Persistenza di Dati: ORM (Object-Relational-Mapping): trasforma le tabelle di database relazionali in oggetti utilizzabili all’interno del codice. (JPA, EF, ecc)

Rappresentazione dei dati: XML oppure JSON

• XML (eXtensible Markup Language): linguaggio di markup estendibile in quanto permette di creare tag personalinnati.

• JSON (JavaScript Object Notation): formato adatto per lo scambio dei dati in applicazione client-server.

Che cos’è Spring MVC?

Framework per realizzare applicazioni web bste sul modello MVC sfruttando Inversion of Control(tramite dependency injection) e aspect oriented programming.

Model: classi che rappresentano le entità gestite dall’applicazione.  
View: rappresentano I vari file che vengono compilati in HTML o da altri tipi di rappresentazioni delle entità.  
Controller: Sono le classe che si occupano di gestire la richiesta dell’utente.

**Che cos’è Spring Framework?**

Framework open source per lo sviluppo di applicazioni di tipo Enterprise.

Principali Caratteristiche:

• Dependency Injection

• AOP

• Spring MVC web application / Restful Web Service framework

• Foundational support per JDBC, JPA, JMS

**Quali sono I vantaggi di HTML5 riguardo HTML ?**

Nuove elementi, cancellazione di elementi obsoleti. Riduzione dei plugin esterni, miglior gestione di errori, scripting sostituito da markup, sviluppo linguaggio aperto e pubblico.

**Che cosa serve CSS?**

Serve a definire gli aspetti visuali (presentazione) di una pagina.

Vantaggi

Separanno i contenuti dalla presentazione, di modo a ottenere un codice HTML piu lineare e pulito. Riduce il peso della pagina e quindi i tempi di caricamento.

**Cosa è lo Spring DispatcherServlet?**

Gestisce le richieste e le risposte HTTP, comunicando tra l’handler mapping, controller, view e view resolver.

**Cosa è Bootstrap?**

È una raccolta di strumenti per la creazione di siti e applicazioni web. Contiene modelli di progettazione basati su HTML, CSS e JS.

**Cosa è SASS?**  
È uno scripting language che è interpretato o compilato in CSS. Offre diverse possibilità tra cui: variabili, nesting, file parziali, imports, ereditarietà, mixins e operatori.

**Cosa sono Cookie?**Sono stringhe di testo di piccole dimensione (4kb) inviate da un server ad un web client e rimandate indietro al server per accedere allo stesso dominio web.

**Come sono composti I Cookie?**  
Sono composti da nome/valore. Hanno una scadenza (di default chiusare browser), possono essere HttpOnly (rende invisibile a javascript e linguaggi client-side), sicuro (https) e dominio e percorso.

**Quali sono I tipi di cookie?**

* Session cookie (solo in memoria temporanea e cancellati all’uscita del browser)
* Persistent cookie (Dura di piu della sessione dell’utente, non è cancellato una volta usciti)
* Secure cookie (Solo usato via HTTPS, cookie sempre criptato)
* HttpOnly Cookie (usato solo quando si trasmettono richieste http(s) limitando il cross-site s.)

**Quali sono le alternative al cookie?**Tracking dell’IP, URL (query string), Hidden form fields (il server nasconde le informazioni in campi hidden), storage apposite sul browser (sessionStorage, localStorage, webSql, ….)

**Cosa è una Session?**  
Una sessione serve per tenere traccia degli utenti durante la navigazione su un determinato sito web. Inizia solitamente caricando un applicazione web sul browser e termina chiudendo il browser (ma non solo).

**Cosa è l’autenticazione?**Processo di generazione, trasmissione e verifica di credenziali per stabilire l’identità delle parti

**Cosa è l’autorizzazione?**  
Indica il permesso di accedere alle risorse da parte degli utenti.

Usabilità

**Cos’è usabilità?**

Il grado in cui un prodotto può essere usato da particolari utenti per raggiungere certi obiettivi con efficacia(Raggiungere l’obiettivo con accuratezza e completezza), efficienza(usare bene le risorse) e soddisfazione( il comfort e l'accettabilità del sistema di lavoro per i suoi utenti e le altre persone influenzate dal suo uso).

**Cosa sono I modelli mentali di Norman?**

Modello Mentale: é il modello mentale creato dall’utente quando interagisce con un sistema/dispositivo.

Modello progettuale: Modelli concettuale del sistemi costruiti dai progettisti.

Immagine del sistema: Risulta dalla struttura fisica che é stata construita (parte visibile)

**Accessibilità Web**

**Cos’è l’accessibilità?**

Le persone con disabilità possono usare il Web.  
Il Web Accessibility Initiative definisce linee guida per rendere il Web accessibile a queste persone.

**Quali sono le disabilità?**

* Visive
* Uditive
* Fisiche
* Cognitive
* Del linguaggio

**Cosa sono le tecnologie assistive?**

HW e SW realizzati per rendere accessibli prodotti informatici

**Linee guida per l’accessibilità definite da W3C:**

* WCAG (Web Content Accessibility Guidelines): riguarda pagine e applicazioni web, Percepibile, Operabile, Comprensibile, Robusto
* UAAG (User Agent Accessibility Guidelines): riguarda browsers web
* ATAG (Authoring Tool Accessibility Guidelines): riguarda tool usati per la creazione di pagine web
* WAI-ARIA (Accessible Rich Internet Applications): riguarda pagine dinamiche e web app

**Che cos’è WAI-ARIA?**

Sono delle linee guida che forniscono un framework per aggiungere attributi che identificano caratteristiche per l’interazione utente, relazionarli tra loro e conoscere il loro stato.  
È un insieme di attributi specifici per accessibilità che possono essere aggiunti a dei markup, HTML.

I ruoli identificati si dividono in 4 categorie:

* Abstract roles
* Widget roles
* Document Structure Roles
* Landmark Roles

**Quali sono gli stati e le proprietà di WAI-ARIA?**

* Stati: aggiungono informazioni dinamiche sullo stato attuale di un elemento
* Live Regions: descrive il tipo di aggiornamento che riceve l’elemento
* Landmarks: usate per identificare aree distinte nella pagina, in tre livelli (navigation della pagina, sidebar, main content)

Quando bisognerebbe usare WAI-ARIA?

* Indicatori/riferimenti
* Aggiornamento dinamico del contenuto
* Migliorare l’accessibilità da tastiera
* Accessibilità dei controlli non semantici

Usare WAI-ARIA solo quando è necessario! Usare sempre funzionalità HTML native

**Definizioni**

**Definizione di contenuto:**

Oggetto che contiene o cattua informazione attaverso diversi formati. Consideriamo come Contenuto Digitale , file o stream di dati elaborati in forma di informazione.

**Definizione di informazione:**

Risultato dell’interpretazione dei dati.

**Definizione di Dati:**

Flusso di valori concernenti fatti che rappresentano eventi che accadono nel mondo reale prima che siano organizzati in una forma che possa essere commpresa e utilizzata da un essere umano.

**Definizione di Conoscenza:**

Comprensione ed interpretazione di informazioni e dati che rappresenta un modo operativo esplicitabile e trasmettibile.

**Definizione di Competenza:**

Modo operativo non esplicitabile e non direttamente trasmettivile. Relazionato alla performance di attività.

*Esempio:*

*Dati → Database*

*Informazioni → Pagine Web o Applicazioni*

*Conoscenza → Knowledge Base, Web semantico e XML.*