

Sistemi informativi su Web

PRESENTAZIONE DEL CORSO

aa 2024-2025



Paolo Merialdo
Università degli Studi Roma Tre



This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States
See <http://creativecommons.org/licenses/by-nc-sa/3.0/us/> for details



Sistemi Informativi su Web

Presentazione del corso (aa 2024-2025)

Paolo Merialdo
paolo.merialdo@uniroma3.it

Docente

- Paolo Merialdo
 - <http://merialdo.dia.uniroma3.it>
 - paolo.merialdo@uniroma3.it
(alias paolo.merialdo@gmail.com)
 - Via della Vasca Navale 79, stanza 2.13
 - Orario ricevimento studenti:
dopo la lezione oppure lun alle 12.45
 - Materiale didattico e comunicazioni su Moodle

Contenuti del corso

- Tecnologie e metodologie per la progettazione e la realizzazione di Sistemi informativi su Web
- Il corso affronta 3 principali argomenti
 - Tecnologie e metodologie per la gestione della persistenza
 - Tecnologie e metodologie Web lato server
 - Tecnologie e metodologie Web lato client
- Tecnologie: HTML, CSS, Java EE, Spring Boot, JavaScript, RDBMS

Difficoltà

- Sistemistiche
 - installare e far funzionare diversi componenti
- Tecnologiche
 - capire e imparare ad usare (insieme) diversi linguaggi, librerie, sistemi
- Metodologiche
 - capire e imparare come applicare concetti e metodi,
 - capire e imparare come progettare e implementare SIW con gli strumenti proposti

Prerequisiti

- Programmazione OO
 - Tutto (ma proprio tutto)
- Basi di dati
 - Modello relazionale
 - Progettazione concettuale e logica
 - SQL: query "semplici (SPJ)", aggiornamenti
- Reti di calcolatori
 - Protocolli
 - Architetture

Organizzazione didattica

- Lezioni frontali (ca 35h)
- Laboratorio (ca 20h)
 - In aula con il calcolatore
- E' un corso molto pratico: useremo molto spesso il calcolatore
 - quindi portatelo sempre, con la batteria bella carica

Materiale didattico

- Copia delle trasparenze e dei sorgenti sw
 - Sul sito web del corso
 - In genere, le trasparenze sono pubblicate prima della lezione, ma possono essere aggiornate subito dopo la lezione
- Riferimenti web
 - Articoli e documentazione disponibile su Web segnalati dal docente

Esame

- 2 progetti (obbligatori)
 - 1 proposto dal docente
 - 1 ideato dallo studente
- Orale
 - discussione sul progetto
- Voto: orale 20 punti + progetti 12 punti
- Attenzione: ad ogni appello il progetto assegnato dal docente cambia

Progetto (appelli giugno/luglio)

- Progetto proposto dal docente
 - le specifiche sono fornite circa un mese prima dell'esame
 - il progetto deve essere svolto **individualmente**
 - tecnologie: quelle proposte nel corso
- Progetto proposto dallo studente
 - gli studenti propongono e discutono con il docente la propria idea di progetto verso la metà di aprile
 - per gli studenti frequentanti che sostengono il progetto deve essere svolto in **gruppi** di 2-3 persone
 - tecnologie: quelle proposte nel corso + altre tecnologie (proposte dal docente) su cui il gruppo lavora autonomamente

Progetto (appelli settembre/febbraio)

- Progetto proposto dal docente
 - le specifiche sono fornite circa un mese prima dell'esame
 - il progetto deve essere svolto **individualmente**
 - tecnologie: quelle proposte nel corso
- Progetto proposto dallo studente
 - gli studenti propongono e discutono con il docente la propria idea di progetto circa un mese prima dell'esame
 - il progetto deve essere svolto **individualmente**
 - tecnologie: quelle proposte nel corso + altre tecnologie (proposte dal docente) su cui il gruppo lavora autonomamente

Le modalità di esame sono chiare?

Regole generali

- Le lezioni iniziano alle 8:30 e finiscono alle 10.00
- **Fate domande**
- **Fate domande**
- **Fate domande**
- Usate Google e/o un LLM (non reinventiamo la ruota), ma con la testa!

Scelte didattiche

- Lato back-end, il corso approfondisce le vostre conoscenze Java
 - Spring Boot framework
- Vedremo una parte (limitata, ma significativa e didatticamente fondamentale) di queste tecnologie
- Perché una piattaforma Java
 - non esiste LA soluzione (vedi slide successiva)
 - è (ancora) molto usata, soprattutto in ambito enterprise
 - ci permette di concentrarci sugli aspetti metodologici, che sono comuni a tutte le piattaforme (vedi slide successiva)

Piattaforme alternative

- JSF
 - Framework MVC
 - Linguaggio Java
- Ruby on Rails
 - Framework MVC
 - Linguaggio Ruby
- Django
 - Framework MVC
 - Linguaggio Python
- Laravel
 - Framework MVC
 - Linguaggio PHP
- Express js
 - Framework MVC
 - Linguaggio JavaScript
- Genie
 - Framework MVC
 - Linguaggio Julia

https://en.wikipedia.org/wiki/Comparison_of_web_frameworks

Scelte didattiche

- Lato front-end, il corso introduce concetti base
 - HTML
 - CSS
 - JavaScript
- Vedremo una parte (molto limitata, ma didatticamente rilevante) di un framework front-end
- Perché poco spazio al front-end?
 - abbiamo 6 cfu
 - i framework front-end hanno un'evoluzione tumultuosa (da un paio di anni hanno iniziato a stabilizzarsi)

L'elefante nella stanza



Un progetto sw (reale) implica l'uso di molteplici strumenti

– Ide

- Useremo Eclipse (e una sua variante, STS)
- Ma potete usare IntelliJ o Visual Studio Code

– Version control system

- Useremo git (con github)

– Sistema di gestione di progetti

- Useremo Maven
- Ma potete usare Gradle

– Sistema di logging

- Useremo Log4j2 e SLF4J

Un progetto sw (reale) implica l'uso di molteplici tecnologie

- Linguaggi di programmazione
 - Java, Javascript, SQL
(come minimo)
- Linguaggi di marcatura e presentazione
 - HTML
 - CSS
- DBMS
 - Usiamo Postgres
 - Ma potete usare MySql, SQLite, MariaDb
- Librerie
 - Gestione della persistenza
 - Gestione della sicurezza
 - Gestione della validazione delle form
 - Gestione delle viste

Un Ingegnere informatico consapevole

- Ha chiari i concetti chiave
- Sa governare i dettagli sistemistici
- Sa scegliere le tecnologie più adatte al contesto
- Impara ad approfondire autonomamente e criticamente
- Sa usare un LLM per diventare più produttivo (non per farsi sostituire: verresti sostituito da un ingegnere più consapevole)