UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Technological Infrastructures for Data Science

2627-2-FDS02Q016

Obiettivi

Conoscenza e capacità di comprensione

Al termine del corso, lo studente avrà acquisito:

- Conoscenze fondative delle principali soluzioni tecnologiche a supporto della data science, includendo virtualizzazione, cloud computing e containerization
- Comprensione dei modelli architetturali di riferimento per le infrastrutture di data science, con particolare focus su Data Lake, HDFS e YARN
- Conoscenze teoriche sui paradigmi di processamento dati (batch processing, stream processing, messaging) e relative tecnologie (Hadoop, Spark, Storm, Kafka)
- Comprensione delle metodologie di sviluppo software moderne (Waterfall, Agile, DevOps, DataOps, MLOps) nel contesto della data science
- Conoscenze di base per analizzare e valutare la possibilità di applicare soluzioni infrastrutturali esistenti a problemi specifici di data science

Conoscenza e capacità di comprensione applicate

Al completamento del corso, lo studente sarà in grado di:

- Progettare e implementare soluzioni di containerization utilizzando Docker per applicazioni di data science
- Utilizzare strumenti e piattaforme cloud per la gestione e l'elaborazione di grandi volumi di dati
- Configurare e utilizzare sistemi distribuiti per il batch e stream processing di dati
- Applicare metodologie di sviluppo software appropriate ai progetti di data science e machine learning
- Sperimentare con soluzioni tecniche e tecnologiche specifiche attraverso attività laboratoriali pratiche
- Interagire efficacemente con gli strumenti tecnologici presentati durante il corso

Competenze trasversali

Il corso contribuisce allo sviluppo delle seguenti competenze trasversali:

Autonomia di giudizio: attraverso l'analisi critica di diverse soluzioni tecnologiche e la valutazione della loro applicabilità a contesti specifici di data science.

Abilità comunicative: mediante la discussione orale di approfondimenti su tematiche del corso e la capacità di presentare soluzioni tecniche in modo chiaro e strutturato.

Capacità di apprendere: fornendo gli strumenti metodologici per rimanere aggiornati sull'evoluzione delle tecnologie per la data science e per approfondire autonomamente nuove soluzioni infrastrutturali.

Contenuti sintetici

Sezioni Tematiche

Il corso è costituito dai seguenti moduli:

Modulo 1 - Infrastruttura: Introduzione alla virtualizzazione, Cloud Computing e Containerization

Modulo 2 - Piattaforma: Organizzazione e distribuzione dei dati, Data Lake, HDFS, YARN

Modulo 3 - Processamento: Batch vs Streaming vs Messaging, i casi di Hadoop, Spark, Storm, Kafka

Modulo 4 - Sviluppo software: Waterfall, Agile, DevOps, DataOps, MLOps

Programma esteso

Argomenti del corso divisi per moduli:

Modulo 1 - Infrastruttura

- La figura del data engineer
- L'architettura di riferimento
- Virtualizzazione
- Cloud Computing (Introduzione, Modelli di servizio e di deployment, cararatteristiche essenziali)
- Containerization con Docker
- Serverless

Modulo 2 - Piattaforma

- Il Data Lake
- HDFS e YARN

Modulo 3 - Processamento

- Batch processing (Apache Hadoop e Apache Spark)
- Stream processing (Apache Storm, Apache Spark e Apache Flink)

- Messaging (Apache Kafka)

Modulo 4 - Sviluppo Software

- Service computing
- Software engineering
- Metodologie di sviluppo (Waterfall, Agile, DevOps, DataOps, MLOps)

Prerequisiti

Conoscenze tecniche fondamentali

- **Architettura dei sistemi informatici**: comprensione dell'organizzazione e del funzionamento dei componenti hardware (processore, memoria RAM, dispositivi di archiviazione) e della loro interazione
- **Sistemi operativi**: familiarità con i concetti base e l'utilizzo pratico di ambienti Unix/Linux o Windows
- **Interfaccia a riga di comando**: competenza nell'uso della shell/terminale per navigazione del filesystem, gestione file e esecuzione di comandi base

Competenze di programmazione

- **Linguaggio Python**: conoscenza delle strutture dati fondamentali (liste, dizionari, tuple), costrutti di controllo (cicli, condizioni), funzioni e gestione base di librerie
- **Ambiente Jupyter**: esperienza nell'utilizzo di Jupyter Notebook/Lab per sviluppo interattivo, visualizzazione dati e documentazione del codice

Livello richiesto

Le competenze elencate dovrebbero corrispondere a quelle acquisite in corsi introduttivi di informatica o programmazione a livello undergraduate, o attraverso esperienza pratica equivalente.

Modalità didattica

Il corso adotta un approccio didattico integrato che combina diverse metodologie per favorire l'apprendimento attivo e il coinvolgimento degli studenti:

Articolazione delle attività

Lezioni frontali interattive

- Circa 15 lezioni da 2-3 ore ciascuna svolte in modalità interattiva in presenza
- Durante le lezioni il docente promuoverà la partecipazione attiva degli studenti attraverso domande, discussioni guidate, analisi di casi studio e momenti di riflessione condivisa

- Le lezioni integreranno contenuti teorici con esempi pratici e applicazioni concrete

Laboratori pratici

- 6 laboratori da 3 ore ciascuno (oppure 9 da 2 ore) svolti in presenza con modalità interattiva
- Attività hands-on per l'applicazione pratica dei concetti teorici
- Lavoro individuale e di gruppo sotto la supervisione del docente

Contenuti asincroni

- Una quota di circa 10% del corso verrà erogata in modalità asincrona mediante video-registrazioni (4-5 ore)
- I contenuti asincroni integrano le lezioni frontali
- Gli studenti potranno accedere a questi contenuti secondo i propri tempi di studio per consolidare e ampliare gli argomenti trattati in aula

Lingua di erogazione

Il corso verrà interamente erogato in **lingua inglese**

Materiale didattico

Dispense e slide del corso fornite dai docenti.

Si segnalano i seguenti testi per approfondimento:

- The basics of cloud computing ISBN-13: 978-0124059320 Autori: Derrick Rountree, Ileana Castrillo
- Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems ISBN-13: 978-1449373320 Autore: Martin Kleppmann

Periodo di erogazione dell'insegnamento

Secondo anno, primo semestre

Modalità di verifica del profitto e valutazione

La valutazione dell'apprendimento si articola in **due componenti complementari**, entrambe obbligatorie:

Prova scritta (28 punti)

La prova scritta valuta la conoscenza degli argomenti trattati durante il corso attraverso:

- **13-15 domande a scelta multipla** per verificare la comprensione dei concetti fondamentali
- **2-3 domande a risposta aperta** per valutare la capacità di analisi e sintesi

Durata: 60-75 minuti

Punteggio massimo: 28 punti

Prova orale - Presentazione di approfondimento (4 punti)

La prova orale consiste nella presentazione e discussione di un tema di approfondimento scelto tra:

- Argomenti correlati al corso ma non trattati durante le lezioni
- Approfondimenti di tematiche già affrontate in aula

Modalità di svolgimento:

- **Preparazione**: il lavoro di ricerca e la creazione delle slide possono essere realizzati in gruppi di 2 persone
- **Presentazione e valutazione**: individuali e personalizzate per ciascun componente del gruppo
- **Approvazione del tema**: l'argomento deve essere previamente concordato con il docente

Punteggio massimo: 4 punti

Criteri di superamento dell'esame

L'esame si considera **superato** quando si verificano **entrambe** le seguenti condizioni:

1. **Soglia minima per ciascuna prova**: punteggio ? 16 per la prova scritta e ? 2 in quella orale
2. **Punteggio complessivo**: somma dei punteggi delle due prove ? 18 punti

Voto finale: corrisponde alla somma dei punteggi delle due prove (massimo 32 punti, corrispondente a 30 e lode).

Note importanti

- **Non sono previste prove parziali** durante lo svolgimento del corso
- Il mancato raggiungimento della soglia minima in una delle due prove comporta la ripetizione della specifica prova e non dell'intero esame

Orario di ricevimento

Martedì 12:30-14:30, chiedere conferma per email

Sustainable Development Goals

IMPRESE, INNOVAZIONE E INFRASTRUTTURE