Università degli Studi di Napoli Federico II

CORSO DI LAUREA MAGISTRALE in INGEGNERIA INFORMATICA

Scuola Politecnica e delle Scienze di Base

Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

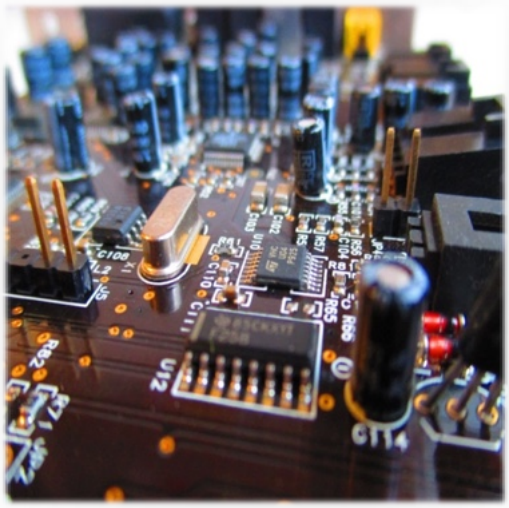
Classe delle Lauree Magistrali in Ingegneria Informatica LM-32
Matricola DE9

Il corso di studio in breve

Obiettivo della Laurea Magistrale in Ingegneria Informatica è formare un professionista in grado di inserirsi in realtà produttive molto differenziate e in rapida evoluzione, nei settori della progettazione, ingegnerizzazione, sviluppo, manutenzione di servizi, applicazioni ed impianti informatici, di sistemi informativi aziendali complessi.
La formazione professionale del laureato magistrale in Ingegneria Informatica richiede l'acquisizione di capacità progettuali avanzate e con contenuti innovativi nell'area dei sistemi di elaborazione.
Il corso di laurea magistrale in Ingegneria Informatica è organizzato nelle seguenti aree tematiche.
Una prima area tematica di interesse formativo è quella relativa alla **Data Engineering and Artificial Intelligence**. L'obiettivo di questa area tematica è la formazione di ingegneri esperti nella progettazione sistemi informatici complessi in grado di gestire in modo efficiente grandi quantità di dati ed estrarre conoscenza "utile" da questi ultimi attraverso avanzate tecniche di analytics.
Ulteriore area tematica è quella relativa ai **Embedded Systems for Industry and IoT**. I sistemi embedded sono sistemi di elaborazione delle informazioni, destinati ad eseguire delle specifiche applicazioni. La loro diffusione è in costante aumento, essi rappresentano il cuore delle moderne applicazioni industriali e sono presenti in numerosissimi settori applicativi: sistemi automobilistici, telemedicina, dispositivi multimediali, sistemi robotici, domotica, Internet of Things, dispostivi per la gestione della sicurezza, sistemi real time, sistemi per il controllo di infrastrutture e sistemi di elaborazione per applicazioni di AI.
Internet and cloud costituisce una area tematica che ha l'obiettivo di fornire agli studenti competenze su tecnologie avanzate in ambito networking al fine di formare professionisti in grado di maneggiare il complesso insieme di protocolli, architetture e applicazioni in uso correntemente e in futuro.
Un'ulteriore area tematica di interesse per il percorso formativo dell'Ingegnere Informatico magistrale è **Intelligent Robotic Systems,** legata alla robotica intelligente, che rappresenta una tecnologia emergente in rapida crescita con un altissimo potenziale innovativo e una varietà di applicazioni in ambito industriale, domestico, e di servizio, in ambito logistico, in ambito sanitario, nei sistemi di sorveglianza, ecc.
Infine, il tema della **Cyber-Security** ricopre una rilevanza strategica nel mondo dell'Information Technology, sia in ambito nazionale che internazionale. Obiettivo di questa area tematica è quello di formare i nuovi "ingegneri della sicurezza", ovvero figure professionali competenti nella protezione dei sistemi, del software e delle reti.

Percorsi curricolari

La formazione professionale del laureato magistrale in Ingegneria Informatica richiede l'acquisizione di capacità progettuali avanzate e con contenuti innovativi nell'area di **Internet and Cloud**, in quella degli **Embedded Systems for Industry and IoT**, in quella di **Data Engineering and Artificial Intelligence**, in quella della **Cyber-Security** e in quella di **Intelligent Robotic System**.



L'area tematica **Internet and cloud** ha l'obiettivo di fornire agli studenti competenze su tecnologie avanzate in ambito networking, al fine di formare professionisti in grado di maneggiare il complesso insieme di protocolli, architetture e applicazioni in uso correntemente e in futuro (con riferimento ai sistemi cloud, alle reti wireless moderne, al paradigma infrastructure-as-code). In questo ambito sono di particolare interesse i principi per la progettazione e la gestione delle reti, tenendo in conto le problematiche legate alla gestione delle risorse, l'instradamento del traffico e la robustezza a fallimenti ed attacchi, sia in reti tradizionali che in reti avanzate.

L'area tematica **Embedded Systems for Industry and IoT** si concentra sulle metodologie e le tecnologie per il progetto e la realizzazione dei sistemi di elaborazione cosiddetti "dedicati", la cui diffusione è in costante aumento e che rappresentano il cuore delle moderne applicazioni dell'Industria 4.0 e di numerosissimi altri settori e sistemi (sistemi di bordo dei veicoli, telemedicina, dispositivi multimediali, sistemi robotici, Internet delle Cose (IoT), sistemi di supervisione e controllo delle infrastrutture, ecc.). In questo ambito viene dato particolare risalto ai requisiti di prestazioni, scalabilità, affidabilità, ed elaborazione real-time, e alle architetture avanzate per il calcolo ad alte prestazioni.

Percorsi curricolari

L'area tematica **Data Engineering and Artificial Intelligence** ha l'obiettivo di formare ingegneri esperti nella progettazione e manutenzione di sistemi informatici complessi in grado di gestire in modo efficiente, attraverso apposite infrastrutture hardware/software, grandi quantità di dati ed estrarre conoscenza "utile" da questi ultimi, attraverso le più avanzate tecniche di analytics di tipo "data-centric". Nel contempo, vengono forniti strumenti ormai imprescindibili nell'ambito dei moderni sistemi informatici, a supporto dei processi operativi e direzionali di qualsiasi organizzazione. Vengono quindi affrontate nel dettaglio le problematiche relative alla realizzazione di sistemi "intelligenti", basati sulle più recenti metodologie e tecniche dell'Artificial Intelligence (AI), dal machine/deep learning ai sistemi cognitivi, dall'information retrieval alla visione artificiale, nonché quelle relative alla loro messa in produzione e testing in accordo al paradigma MLOps ed agli aspetti legati all'etica dei modelli di AI nei differenti contesti applicativi.

L'area tematica **Cyber-Security** ha l'obiettivo di formare i nuovi "ingegneri della sicurezza", ovvero figure professionali competenti nella protezione dei sistemi, del software e delle reti. In questo ambito, che ha ormai una rilevanza strategica sia a livello nazionale che internazionale, alcuni dei profili più richiesti sono i seguenti: il Security Administrator – che ha il compito di rendere operative le soluzioni tecnologiche di security; il Security Architect – che grazie alle sue competenze modellistiche, disegna le misure di sicurezza e le policy adottate dall'organizzazione; il Security Engineer – che ha un bagaglio tecnico e modellistico, si occupa di monitorare i sistemi e proporre soluzioni relative alla risposta agli incidenti; il Security Analyst – che ha competenze di analisi di processo e si occupa di valutare le vulnerabilità che possono interessare reti, apparati, applicazioni e servizi proponendo soluzioni ed accorgimenti pratici. La tematica è dunque complessa e richiede professionisti altamente qualificati con competenze specialistiche nel campo dell'ingegneria dei sistemi, dell'ingegneria del software e delle reti di comunicazione.

Percorsi curricolari

L'area tematica **Intelligent Robotic Systems** fornisce agli studenti di Ingegneria Informatica le competenze metodologiche di base necessarie per la modellazione, pianificazione e controllo di sistemi robotici, e gli strumenti hardware e software per lo sviluppo di applicazioni robotiche intelligenti. Verranno approfondite le tematiche relative ai sistemi operativi real time, alla progettazione di sistemi per l'acquisizione ed elaborazione di informazioni sensoriali (ad es. implementazione di driver per l'interfacciamento con i dispositivi), alla configurazione ed utilizzo di algoritmi di machine learning per l'estrazione e rappresentazione della conoscenza e per la pianificazione delle azioni dei robot in autonomia. L'ingegnere informatico con una formazione in questa area sarà in grado di progettare e realizzare sistemi robotici intelligenti in grado di supportare complessi task cognitivi, con riferimento ai sistemi di sensori, ai sistemi di controllo e agli algoritmi per il supporto alle decisioni, interagendo efficacemente con gli esperti nei diversi settori applicativi

iStock
Credit: ipopba

Sbocchi occupazionali

Con una crescente richiesta di competenze ingegneristico/informatiche, la specifica preparazione del laureato magistrale in Ingegneria Informatico lo rende uno dei professionisti più ricercati sul mercato del lavoro.
La maggior parte dei laureati in Ingegneria Informatica viene assorbita dalle numerose aziende di servizi e dalle industrie del territorio nazionale ed internazionale.
Fra gli sbocchi occupazionali si citano i seguenti:
- il settore dei servizi pubblici e privati;
- le aziende di sevizi di consulenza;
- le imprese produttrici e fornitrici di apparecchiature e sistemi informatici e industriali;
- le aziende che sviluppano prodotti e servizi ad alto contenuto informatico;
- le pubbliche amministrazioni;
- l'industria in genere.

Modalità d'ammissione

I requisiti curriculari richiesti includono il conseguimento di una Laurea nella classe L-8. Inoltre, l'accesso è subordinato alla verifica dei requisiti curricolari e dell'adeguatezza della preparazione dello studente. Ulteriori informazioni sono riportate al seguente link: LM_informatica

Attività di tirocinio curriculare

Il corso di studi prevede 3 CFU di "**Altre Attività Formative**" e 9 CFU di "**Stage o tirocini presso imprese**" che possono essere acquisite mediante tirocini extramoenia o intramoenia.
Il tirocinio extramoenia è svolto presso aziende, centri di ricerca o altri enti pubblici e/o privati, italiani o esteri, con affiancamento di un tutor dell'azienda o dell'ente e la supervisione di un tutor universitario. Il tirocinio intramoenia è svolto presso laboratori di ricerca dell'ateneo con affiancamento di un tutor universitario (docente o ricercatore).

Prova finale

La Laurea magistrale in Ingegneria Informatica si consegue dopo aver sostenuto una prova finale alla quale si viene ammessi una volta conseguiti tutti i crediti previsti dal proprio piano di studi, esclusi quelli relativi alla prova finale stessa.
La commissione didattica del corso di studio ha predisposto un sistema automatico per supportare gli studenti nella scelta, nella gestione e nella stesura finale della prova finale (tutte le informazioni sono reperibili al seguente link: esame di laurea magistrale).
La prova finale consiste nella valutazione della Tesi di Laurea Magistrale, consistente in una relazione scritta (eventualmente in lingua inglese) avente per oggetto un progetto originale sviluppato dallo studente in modo autonomo sotto la guida di un relatore e eventuali co-relatori. Il documento che presenta le linee guida per l'assegnazione del voto di laurea magistrale è pubblicato sul sito del corso di laurea.
La prova prevede la discussione dell'elaborato di laurea, che verte su attività formative svolte nell'ambito di uno o più insegnamenti ovvero di attività di tirocinio. Al candidato è consentito di avvalersi di un supporto audio-visivo, da proiettare pubblicamente, oppure, in alternativa, di redigere un fascicoletto di sintesi, da consegnare in copia a ciascun componente della Commissione.
Al termine della presentazione, ciascun docente può rivolgere osservazioni al candidato, inerenti all'argomento del lavoro di tesi. La presentazione ha una durata compresa di norma in 10 minuti.

Requisiti curriculari minimi per l'accesso alla Laurea Magistrale in Ingegneria Informatica (LM-32)

Lo studente in possesso del titolo di Laurea ex D.M. 509/99 o ex D.M. 270/04 potrà essere ammesso al Corso di Laurea Magistrale in Ingegneria Informatica se avrà acquisito nella precedente carriera CFU nei settori scientifico disciplinari di seguito indicati nella misura minima corrispondentemente indicata:

SS	CF minim
MAT/05, MAT/03, MAT/09, FIS/03,	2
IN -INF/01, -INF/02, -INF/03, -INF/04 IN -INF/05, -INF/06, -INF/07, -IND/13 IN -IND/16, -IND/17, -IND/31, -IND/32 IN -IND/34, -IND/35,	5

Referenti

Coordinatore Didattico del Corso di Studi in Ingegneria Informatica:
Prof. Simon Pietro Romano
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione
E-mail: spromano@unina.it

Referente del Corso di Studi per il Programma ERASMUS+:
Prof. Alessio Botta
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione –
E-mail: a.botta@unina.it

Referente del Corso di Studi per i tirocini:
Prof. Antonio Montieri
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione
E-mail: antonio.montieri@unina.it

Referente del Corso di Studi per l'Orientamento:
Prof. Michela Gravina
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione
E-mail: michela.gravina@unina.it

Piano di studi

Per vedere tutte le informazioni sul piano di studi, gli assegnamenti delle cattedre e ulteriori informazioni visitare il seguente link.

Insegnamenti primo anno di corso

Denominazione insegnamento	Sem.	CFU	SSD	Ambito Disciplinare	Tipo(*)
Architettura dei Sistemi Digitali	I	6	ING-INF/05	Ingegneria Informatica	B
Algorithms, Data Structures and Machine Learning: *Modulo di Algorithms and Data Structures*	I	6	ING-INF/05	Ingegneria Informatica	B
Attività formative **affini o integrative** a scelta dello studente (lo studente dovrà scegliere da 0 a 2 insegnamenti dalla **Tabella A**, vedi nota a)	I	0-18		Attività formative affini o integrative	C
Attività formative a **scelta autonoma** dello studente (**Tabella C**, vedi nota b)	I	0-18		A scelta dello studente	D
Architettura e Progetto dei Calcolatori	II	9	ING-INF/05	Ingegneria Informatica	B
Networks and Cloud Infrastructures	II	6	ING-INF/05	Ingegneria Informatica	B
Algorithms, Data Structures and Machine Learning: *Modulo di Machine Learning*	II	6	ING-INF/05	Ingegneria Informatica	B
Attività formative **affini o integrative** a scelta dello studente (lo studente dovrà scegliere da 0 a 2 insegnamenti dalla **Tabella A**, vedi nota a)	II	0-18		Attività formative affini o integrative	C
Attività formative a **scelta autonoma** dello studente (**Tabella C**, vedi nota b)	II	0-18		A scelta dello studente	D

Nota a) - Lo studente può scegliere di usufruire dei **18 CFU** di tipologia C, ovvero **insegnamenti affini o integrativi** riportati in Tabella A, tra il primo e il secondo semestre del **primo anno**.
Nota b) - Lo studente può scegliere di usufruire dei **18 CFU** di tipologia D, ovvero **insegnamenti a scelta autonoma** riportati in Tabella C, al primo e/o al secondo semestre del **primo e/o** del **secondo anno**.

Insegnamenti secondo anno di corso

Denominazione insegnamento	Sem.	CFU	SSD	Ambito Disciplinare	Tipo(*)
Impianti di elaborazione	I	9	ING-INF/05	Ingegneria Informatica	B
Attività formative **curriculari** a scelta dello studente (Lo studente dovrà scegliere al più 3 insegnamenti da una delle aree presenti in **Tabella B**)	I	0-18	ING-INF/05	Ingegneria Informatica	B
Attività formative **a scelta autonoma** dello studente (**Tabella C**, vedi nota b)	I	0-18		A scelta dello studente	D
Attività formative **curriculari** a scelta dello studente (Lo studente dovrà scegliere al più 3 insegnamenti da una delle aree presenti in **Tabella B**)	II	0-18	ING-INF/05	Ingegneria Informatica	B
Attività formative **a scelta autonoma** dello studente (**Tabella C**, vedi nota b)	II	0-18		A scelta dello studente	D
Altre attività formative	II	3		Ulteriori attività formative	F
Attività di laboratorio o tirocini presso imprese	II	9		Stage o tirocini	F
Prova finale	II	12		Ulteriori attività formative	E

Nota b) - Lo studente può scegliere di usufruire dei **18 CFU** di tipologia D, ovvero **insegnamenti a scelta autonoma** riportati in Tabella C, al primo e/o al secondo semestre del **primo e/o** del **secondo anno**.

Tabella A

Attività formative affini o integrative a scelta dello studente

Insegnamento o attività formativa	Sem.	CFU	SSD	Ambito Disciplinare	Tipo(*)	Mutuazioni
Trasmissione dei segnali digitali	I	9	ING-INF/03	Attività formative affini o integrative	C	
Teoria dell'Informazione	I	9	ING-INF/03	Attività formative affini o integrative	C	LM-TLMD
Architetture dei sistemi integrati	I	9	ING-INF/01	Attività formative affini o integrative	C	LM-IELM
Economia ed Organizzazione Aziendale	I	9	ING-IND/35	Attività formative affini o integrative	C	
Algoritmi di Ottimizzazione Combinatoria e su Rete	II	9	MAT/09	Attività formative affini o integrative	C	
Calcolo Scientifico per l'Innovazione Tecnologica	II	9	MAT/08	Attività formative affini o integrative	C	
Elaborazione di segnali multimediali	II	9	ING-INF/03	Attività formative affini o integrative	C	LM-TLMD

*Lo studente può scegliere di usufruire dei **18 CFU** della tabella sopra riportata, per intero al primo semestre e/o al secondo semestre del primo anno*

Tabella B

Attività formative curriculari a scelta dello studente

Area	Insegnamento o Attività formativa	CFU	SSD	Sem.	Ambito Disciplinare	Tipo(*)
Data Engineering and Artificial Intelligence	Information Systems and Business Intelligence	6	ING-INF/05	I	Ingegneria Informatica	B
	AI Systems Engineering	6	ING-INF/05	I	Ingegneria Informatica	B
	Information Retrieval	6	ING-INF/05	II	Ingegneria Informatica	B
	Big Data Engineering	6	ING-INF/05	II	Ingegneria Informatica	B
Embedded Systems for Industry and IoT	Distributed Systems and IoT	6	ING-INF/05	I	Ingegneria Informatica	B
	High Performance and Quantum Computing	6	ING-INF/05	I	Ingegneria Informatica	B
	Real Time Systems and Industrial Applications	6	ING-INF/05	II	Ingegneria Informatica	B
	Embedded Systems	6	ING-INF/05	II	Ingegneria Informatica	B
Internet and Cloud	Web and Real Time Communication Systems	6	ING-INF/05	I	Ingegneria Informatica	B
	Decentralized Applications and Blockchain	6	ING-INF/05	I	Ingegneria Informatica	B
	Wireless Networks and IoT Technologies	6	ING-INF/05	II	Ingegneria Informatica	B
	Cloud Platforms and Infrastructure-as-Code	6	ING-INF/05	II	Ingegneria Informatica	B
Cyber-Security	System Security	6	ING-INF/05	I	Ingegneria Informatica	B
	Network Security	6	ING-INF/05	I	Ingegneria Informatica	B
	Data Analysis and Cybersecurity	6	ING-INF/05	II	Ingegneria Informatica	B
	Software Security	6	ING-INF/05	II	Ingegneria Informatica	B
Intelligent Robotic Systems (nota c)	Robotics: modulo di Robotic Systems	6	ING-INF/04	I	Ingegneria dell'automazione e robotica	B
	Robotics: modulo di Robotic Lab	6	ING-INF/04	II	Ingegneria dell'automazione e robotica	B
	Un insegnamento a scelta tra:					
	Information Systems and Business Intelligence	6	ING-INF/05	I	Ingegneria Informatica	B

	AI Systems Engineering	6	ING-INF/05	I	Ingegneria Informatica	B
	Distributed Systems and IoT	6	ING-INF/05	I	Ingegneria Informatica	B
	High Performance and Quantum Computing	6	ING-INF/05	I	Ingegneria Informatica	B
	Web and Real Time Communication Systems	6	ING-INF/05	I	Ingegneria Informatica	B
	Decentralized Applications and Blockchain	6	ING-INF/05	I	Ingegneria Informatica	B
	System Security	6	ING-INF/05	I	Ingegneria Informatica	B
	Network Security	6	ING-INF/05	I	Ingegneria Informatica	B
	Big Data Engineering	6	ING-INF/05	II	Ingegneria Informatica	B
	Information Retrieval	6	ING-INF/05	II	Ingegneria Informatica	B
	Real Time Systems and Industrial Applications	6	ING-INF/05	II	Ingegneria Informatica	B
	Embedded Systems	6	ING-INF/05	II	Ingegneria Informatica	B
	Wireless Networks and IoT Technologies	6	ING-INF/05	II	Ingegneria Informatica	B
	Cloud Platforms and Infrastructure as-a-Code	6	ING-INF/05	II	Ingegneria Informatica	B
	Data Analysis and Cybersecurity	6	ING-INF/05	II	Ingegneria Informatica	B
	Software Security	6	ING-INF/05	II	Ingegneria Informatica	B

Lo studente dovrà scegliere 3 insegnamenti da una delle aree presenti in ***Tabella B***
Nota c) - Nel caso di scelta dell'area "Intelligent Robotic Systems", lo studente dovrà inserire l'insegnamento integrato di Robotics (modulo di Robotic Systems al I semestre e modulo di Robotic Lab al II semestre) più un insegnamento a scelta tra quelli in tabella, appartenenti alle altre aree

Tabella C

Attività formative disponibili per la scelta autonoma dello studente

Insegnamento o attività formativa	CFU	SSD	Sem.	Ambito Disciplinare	Tipo(*)	Mutuazioni
Software Architecture Design	6	ING-INF/05	I	A scelta dello studente	D	
Software Testing	6	ING-INF/05	I	A scelta dello studente	D	
Image Processing for Computer Vision	9	ING-INF/03	I	A scelta dello studente	D	LM-TLMD
Gestione Aziendale	9	ING-IND/35	II	A scelta dello studente	D	
Business Processes Automation	3	ING-INF/05	II	A scelta dello studente	D	
Safety Critical Systems	3	ING-INF/05	II	A scelta dello studente	D	
Circuiti per DSP	9	ING-INF/01	I	A scelta dello studente	D	LM-IELM
Instrumentation and Measurements for Smart Industry	9	ING-INF/07	II	A scelta dello studente	D	LM-TLMD
Modelli e Algoritmi di Ottimizzazione	9	MAT/09	II	A scelta dello studente	D	LM-TLMD
Security and Privacy	6	INF/01	I	A scelta dello studente	D	LM-SINF
Ingegneria del suono	6	ING-INF/03	I	A scelta dello studente	D	LM-TLMD
Distributed control and cyber-physical systems design (vecchia denominazione: "*Algoritmi distribuiti e progettazione dei sistemi di controllo su rete*")	6	ING-INF/04	II	A scelta dello studente	D	LM-AUT
Bioinformatica	6	ING-INF/06	II	A scelta dello studente	D	LM-BIO
Quantum Information	6	ING-INF/03	I	A scelta dello studente	D	LM-TLMD
Computer Forensics	6	INF/01	II	A scelta dello studente	D	LM-SINF
Risk Assessment	6	ING-INF/05	II	A scelta dello studente	D	
Cognitive Computing Systems	6	ING-INF/05	II	A scelta dello studente	D	

(*) Legenda delle tipologie delle attività formative (TAF):
A = Base (ex 1)
B = Caratterizzanti (ex 2)
C = Affini o integrativi (ex 4)
D = Attività a scelta (ex 3)
E = Prova finale e conoscenze linguistiche (ex 5)
F = Ulteriori attività formative (ex 6 e 7)

TAF	1	2	3	4	5	6	7
rif. DM270/04	Art. 10 comma 1, a)	Art. 10 comma 1, b)	Art. 10 comma 5, a)	Art. 10 comma 5, b)	Art. 10 comma 5, c)	Art. 10 comma 5, d)	Art. 10 comma 5, e)

Calendario delle attività didattiche

	Inizio	Termine	Apertura sessioni esami 2025/2026
1° periodo didattico	14/09/2026	18/12/2026	14/12/2026 e 07/06/2027
Appelli **ordinari** obbligatori previsti	un appello nel mese di gennaio 2027, un appello nel mese di febbraio 2027		
2° periodo didattico	01/03/2027	11/06/2027	
Appelli **ordinari** obbligatori previsti	un appello nel mese di giugno 2027, un appello nel mese di luglio 2027 ed un appello nel mese di settembre 2027		
Appelli di **recupero** obbligatori previsti	un appello nel periodo 01/03/2027 - 20/03/2027, un appello nel periodo 18/10/2027 – 13/11/2027		

Schede degli insegnamenti

Si riportano di seguito le schede di dettaglio degli insegnamenti previsti dal Corso di Laurea Magistrale in Ingegneria Informatica.
Si avvisano gli studenti che i docenti sono comunque tenuti ad aggiornare le schede degli insegnamenti di cui sono titolari sulla propria pagina del sito www.docenti.unina.it nella sezione **Didattica**-> **Schede Insegnamento.**

SCHEDA DELL'INSEGNAMENTO (SI)

"ARCHITETTURA DEI SISTEMI DIGITALI"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ALESSANDRA DE BENEDICTIS, NICOLA MAZZOCCA
TELEFONO:
EMAIL: ALESSANDRA.DEBENEDICTIS@UNINA.IT, NICOLA.MAZZOCCA@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Conoscenza dei principi di progettazione delle reti logiche combinatorie e sequenziali e dell'architettura generale di un calcolatore; conoscenze di programmazione.

OBIETTIVI FORMATIVI

Il corso si pone l'obiettivo di fornire un'impostazione metodologica e tecnologica per il progetto di architetture dei sistemi digitali dedicati e/o programmabili. Il corso prevede di analizzare le tecniche di progetto con riferimento allo sviluppo di: microcontrollori, processori dedicati, unità di I/O, sistemi di interconnessione, unità aritmetiche, unità dedicate ad applicazioni di IoT e multimediali. Le attività sono svolte con riferimento al linguaggio VHDL, mediante l'impiego di simulatori industriali e di sistemi FPGA (con i relativi ambienti di sviluppo), utilizzati per l'implementazione dei casi di studio proposti durante il corso. Sono inoltre presentati i principali elementi per la realizzazione della documentazione e per il testing di sistemi digitali in applicazioni industriali.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e comprendere le problematiche relative al progetto di sistemi digitali dedicati e programmabili, con riferimento alle metodologie generali di progetto e ai vincoli specifici derivanti dagli strumenti di sviluppo, dai linguaggi di programmazione e dalle piattaforme hardware coinvolti. Deve inoltre dimostrare di comprendere le caratteristiche fondamentali di diversi modelli di sistemi digitali e di saper individuare il modello o i modelli più appropriati per risolvere uno specifico problema di progettazione.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di eseguire l'intero ciclo di progettazione e implementazione di sistemi digitali di base (ad es. macchine combinatorie o sequenziali notevoli) ed avanzati (macchine aritmetiche, reti di interconnessione, interfacce di I/O, processori), dalla fase di definizione dell'architettura e del comportamento funzionale alla descrizione mediante linguaggio VHDL, fino all'implementazione su dispositivi hardware programmabili (FPGA), utilizzando strumenti e ambienti di sviluppo di ampio utilizzo industriale.

PROGRAMMA-SYLLABUS

Principi di progetto di sistemi digitali: Sistemi general purpose, special purpose e embedded. Progettazione dei sistemi digitali: aspetti tecnologici, metodologici e ambienti a supporto. Il ciclo di sviluppo di un sistema digitale. I linguaggi HDL per la descrizione dell'hardware e gli ambienti di simulazione.
Richiami e approfondimenti sul progetto di macchine combinatorie: Realizzazione di funzioni booleane mediante circuiti digitali. Parametri e vincoli di progetto: ritardi, deformazioni dei segnali, dissipazioni di potenza. Minimizzazione di funzioni booleane. Progetto di macchine combinatorie notevoli (multiplexers/demultiplexer, decoder/encoder, reti di transcodifica).
Richiami e approfondimenti sul progetto di macchie sequenziali: Modelli di macchine sequenziali. Macchine impulsive, a livelli, sincrone, asincrone. Progetto di macchine sequenziali notevoli (registri, contatori).
Progetto di sistemi complessi: Modelli architetturali per lo sviluppo di sistemi complessi: sistemi di reti sequenziali, architettura a pipeline, problematiche di tempificazione, modello PO/PC, sistemi a controllo cablato e microprogrammato. Principi di progetto di un processore. Il processore MIC-1.
Protocolli e interfacce di comunicazione fra sistemi: Protocolli di comunicazione sincroni e asincroni; progetto di reti di interconnessione molti a molti; progetto e utilizzo di una interfaccia di comunicazione seriale secondo lo standard RS232.
Architettura e progetto di macchine aritmetiche: Architetture parallele e sequenziali per addizionatori, sottrattori e moltiplicatori binari. Applicazione di architetture multiply and accumulate per l'implementazione di reti neurali.
Dispositivi per la sintesi di reti logiche: PAL, PLA, FPGA, ASIC. Approfondimento sull'architettura degli FPGA XIlinx delle famiglie Artix-7 e Spartan 3E.
Testing di sistemi digitali: Problematiche e tecniche di testing di dispositivi hardware.

Linguaggio VHDL e ambiente di sviluppo: Stili di descrizione dei sistemi digitali e principali costrutti del linguaggio a supporto. Linee guida per l'implementazione di macchine combinatorie e sequenziali notevoli. Processo di simulazione, sintesi, implementazione di un design digitale e strumenti a supporto. Timing analysis e strumenti a supporto.

MATERIALE DIDATTICO

Libro di testo: Conte, Mazzeo, Mazzocca, Prinetto. Architettura dei calcolatori. Edizioni CittàStudi. 2014. ISBN: 9788825173642.
Dispense e presentazioni fornite dal docente relative ad argomenti teorici e applicativi.
Manuali e datasheet dei dispositivi utilizzati per la sintesi dei progetti.
Codice VHDL relativo alle esercitazioni svolte in aula.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso prevede circa il 70% di lezioni frontali in cui vengono affrontati gli argomenti teorici, mentre il restante 30% è riservato a lezioni pratiche ed esercitazioni riguardanti lo sviluppo di codice VHDL per l'implementazione di specifiche macchine e l'utilizzo degli ambienti di sviluppo.
La parte applicativa del corso si avvale di strumenti di sviluppo professionali di cui è disponibile una licenza ad uso gratuito e di board di sviluppo (dotate di FPGA e di diversi dispositivi di I/O) che vengono distribuiti agli studenti per l'implementazione dei propri progetti.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	x
solo scritta	
solo orale	
discussione di elaborato progettuale	x
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	x
	Esercizi numerici	

La verifica dell'apprendimento prevede una prova scritta consistente in esercizi di progetto di sistemi digitali e una prova orale orientata alla verifica della comprensione dei concetti teorici del corso e alla discussione di un elaborato.

b) Modalità di valutazione:

SCHEDA DELL'INSEGNAMENTO (SI)

"ALGORITMI, STRUTTURE DATI E MACHINE LEARNING"

SSD ING-INF/05*

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ROBERTO PIETRANTUONO, CARLO SANSONE
TELEFONO: 0817683880, 0817683640
EMAIL: ROBERTO.PIETRANTUONO@UNINA.IT, CARLO.SANSONE@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE): ALGORITMI E STRUTTURE DATI, MACHINE LEARNING
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): I
SEMESTRE (I, II): ANNUALE
CFU: 12

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

L'insegnamento si propone di fornire in primo luogo le nozioni necessarie per la progettazione e l'analisi di algoritmi e strutture dati nello sviluppo delle applicazioni informatiche. Tali nozioni includono i fondamenti teorici e le tecniche avanzate di progettazione ed analisi di algoritmi la cui applicazione spazia su tutti gli aspetti relativi ad un sistema di elaborazione, da quelli hardware a quelli software, dai sistemi operativi alle reti di elaboratori, dalle basi di dati ai sistemi informativi, dai linguaggi di programmazione all'ingegneria del software, dall'interazione uomo-macchina al riconoscimento dei segnali e delle immagini, all'elaborazione multimediale, all'ingegneria della conoscenza, alla robotica e all'intelligenza artificiale ed alla robotica. Con riferimento a quest'ultimo campo, l'insegnamento si propone di presentare nel dettaglio i principali algoritmi di Machine Learning per la soluzione di problemi di classificazione, predizione numerica e clustering, nonché le metodologie di gestione e sviluppo di un processo di Machine Learning, dalla preparazione dei dati alla valutazione dei risultati. L'insegnamento consentirà anche di sviluppare competenze pratiche nella soluzione di problemi reali di classificazione, predizioni numerica o clustering tramite algoritmi di Machine Learning, grazie ad esercitazioni svolte con tool open source e/o commerciali.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Modulo di Algoritmi e Strutture Dati

Al termine del corso, lo studente deve dimostrare di aver acquisito familiarità con una ampia varietà di strutture dati ed algoritmi noti che risolvono problemi di carattere fondamentale, di aver compreso le tecniche per la sintesi di nuovi algoritmi e di padroneggiare i metodi per analizzare la correttezza e la complessità asintotica degli algoritmi.

Modulo di Machine Learning

Al termine del corso, lo studente deve dimostrare di conoscere i principali algoritmi di Machine Learning e Deep Learning e di essere in grado di scegliere l'algoritmo di Machine Learning più adatto a risolvere uno specifico problemi di classificazione e/o predizione numerica e/o clustering, sulla base dei requisiti del problema medesimo. Lo studente deve inoltre dimostrare di essere in grado di scegliere le opportune tecniche di preparazione dei dati e deve conoscere le tecniche necessarie per la valutazione delle prestazioni di algoritmi di Machine Learning e Deep Learning.

Capacità di applicare conoscenza e comprensione

Modulo di Algoritmi e Strutture Dati

Al termine del corso, lo studente deve dimostrare di sapere applicare e combinare le principali tecniche di progettazione, nonché strutture dati avanzate, per la sintesi di algoritmi corretti ed efficienti per la risoluzione di problemi nello sviluppo delle applicazioni informatiche, e di saperne analizzare formalmente la correttezza e la complessità asintotica. Il percorso formativo è orientato a fornire le capacità e gli strumenti necessarie a risolvere problemi nuovi o non familiari negli ampi contesti relativi ai sistemi di elaborazione dell'informazione.

Modulo di Machine Learning

Al termine del corso, lo studente deve dimostrare di essere in grado di risolvere problemi reali di classificazione, predizione numerica o clustering utilizzando algoritmi di Machine Learning o Deep Learning. Lo studente deve anche dimostrare di saper valutare in modo corretto le prestazioni dei sistemi che realizza.

PROGRAMMA-SYLLABUS

Modulo di Algoritmi e Strutture Dati (6 CFU)

- CONCETTI INTRODUTTIVI (0.25 CFU)
  Algoritmi e strutture dati, ricorsione, analisi e progettazione degli algoritmi.

- TECNICHE DI ANALISI E STRUTTURE DATI ELEMENTARI (1 CFU)
  Analisi di correttezza: invariante di ciclo, correttezza di algoritmi ricorsivi.
  Analisi di complessità: analisi asintotica, notazioni O, Ω, Θ; analisi di algoritmi ricorsivi.
  Strutture dati elementari: dizionari; pile e code; code di priorità; liste; tabelle hash e stringhe; alberi binari di ricerca.

- DIVIDE et IMPERA, ORDINAMENTO (1.25 CFU)
  Problemi ed algoritmi comuni. Ordinamento: merge sort, heap sort, quick sort.
  Ordinamento in tempo lineare (counting sort, radix sort, bucket sort), mediane e statistiche d'ordine.

- RICERCA COMBINATORIALE (0.25 CFU)
  Ricerca esaustiva, ricerca combinatoriale; backtracking, pruning della ricerca

- PROGRAMMAZIONE DINAMICA (1 CFU)
  Introduzione alla programmazione dinamica. Ricerca esaustiva vs. ricerca greedy vs. programmazione dinamica. Applicazioni. Problemi ed algoritmi comuni. Problema di string matching, edit distance, longest increasing sequence. Fibonacci. Problema dello zaino. Ulteriori esempi.

- STRUTTURE DATI E TECNICHE DI ANALISI AVANZATE (1 CFU)
  Alberi RB, alberi auto-aggiustanti. Grafi. Rappresentazione, esplorazione in ampiezza e profondità, ordinamento topologico. Applicazioni. Tecniche di analisi di algoritmi avanzate: analisi ammortizzata.

- PROBLEMI ED ALGORITMI COMUNI, ESEMPI APPLICATIVI (1 CFU)
  Problemi di teoria dei numeri (es.: algoritmi DES ed RSA). Problemi su grafi: ricerca di cammini minimi. Algoritmi paralleli, multithreading dinamico. Esempi (Fibonacci, ordinamento). Analisi e confronto con algoritmi sequenziali. Traduttori ed interpreti: analisi lessicale, analisi sintattica, strutture dati usate nei traduttori.

- PROBLEMI INTRATTABILI (0.25 CFU)
  Introduzione a problemi NP ed NP-completi. Riducibilità. Esempi di problemi NP-completi.

Parte Esercitativa: Prevalentemente in C.

Modulo di Machine Learning (6 CFU)

- INTRODUZIONE AL MACHINE LEARNING (0.25 CFU)

- INPUT ED OUTPUT (0.5 CFU)
  Concetti, istanze ed attributi. Rappresentazione della conoscenza.

- METODI DI BASE (1.25 CFU)
  Modelli probabilistici, alberi di decisione, regole di classificazione, modelli lineari, instance-based e multi-instance learning, clustering.

- VALUTAZIONE DELLE PRESTAZIONI (0.5 CFU)
  Training e Testing. CV, LOO, Cost-sensitive classification. ROC. Valutazione di algoritmi di predizione numerica.

- METODI AVANZATI (2 CFU)
  Alberi di decisione: C4.5. Regole di classificazione. Instance-based learning. Estensione dei modelli lineari: SVM. Epsilon-SVR. MLP. Predizione numerica con modelli lineari.

- TRASFORMARE I DATI (0.25 CFU)
  Selezione degli attributi, PCA, Discretizzazione, Campionamento. One-class classification.

- METODI PROBABILISTICI (0.25 CFU)
  Reti bayesiane. Probability Density Estimation e Clustering. Modelli sequenziali e temporali (cenni).

- DEEP LEARNING (0.5 CFU)
  Addestramento e valutazione delle prestazioni di reti deep, Convolutional Neural Networks, Autoencoders. Reti Neurali ricorrenti e GAN (cenni).

- OLTRE L'APPRENDIMENTO SUPERVISIONATO E NON-SUPERVISIONATO (0.25 CFU)
  Semi-supervised learning. Multi-instance learning.

- ENSEMBLE LEARNING (0.25 CFU)
  Bagging, Randomization, Boosting, Stacking, ECOC.

Parte Esercitativa: Utilizzo di tool open source e/o commerciali (e.g., Knime, Weka).

MATERIALE DIDATTICO

Modulo di Algoritmi e Strutture Dati

LIBRO DI TESTO ADOTTATO

1) Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. **Introduction to Algorithms**. 3rd ed. MIT Press, 2009. ISBN: 9780262033848.

TRASPARENZE DALLE LEZIONI ED ESERCITAZIONI disponibili sul sito web docenti di Ateneo sulla piattaforma Microsoft Teams.

LIBRO CONSIGLIATO

2) Steven Skiena. **The Algorithm Design Manual**, 3rd ed, Springer, 2020. ISBN-13: 978-3030542559, ISBN-10: 3030542556.

Modulo di Machine Learning

LIBRO DI TESTO ADOTTATO

1) Ian H. Witten, Eibe Frank, Mark A. Hall and Christopher J. Pal. **Data mining: practical machine learning tools and techniques**. 4th ed. The Morgan Kaufmann, 2017. ISBN: 9780128042915.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Modulo di Algoritmi e Strutture Dati

Il docente utilizzerà: a) lezioni frontali per circa l'80% delle ore di lezione totali, b) esercitazioni per circa il 20% delle ore di lezione totali. È prevista l'assegnazione di esercizi da svolgere autonomamente e consegnare al docente.

Modulo di Machine Learning

Il docente utilizzerà: a) lezioni frontali per circa il 60% delle ore totali, b) esercitazioni per approfondire aspetti teorici per circa il 35% delle ore totali, c) seminari per circa il 5% delle ore totali.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità d'esame:

L'esame si articola in prova	
scritta e orale	X
solo scritta	
solo orale	
discussione di elaborato progettuale	X
altro	X

In caso di prova scritta i quesiti sono	**A risposta multipla**	
	A risposta libera	X
	Esercizi numerici	

Per il modulo di Algoritmi e strutture Dati, l'esame si articola in n. 2 prove scritte ed in una prova orale che include una discussione sugli esercizi assegnati durante il corso. La parte scritta si articolata in n.1 prova intercorso più n.1 prova finale. Le prove richiedono, per uno o più problemi, di implementare algoritmi per la loro risoluzione ed analizzarne la complessità asintotica.
La prova orale consta di un colloquio sostenuto dopo l'ultima prova scritta. Esso include anche la discussione sulla risoluzione degli esercizi assegnati durante il corso.

Per il modulo di Machine Learning, ciascuno studente dovrà sviluppare durante il corso 3 progetti (c.d. mini-contest), che, in caso di valutazione positiva, lo esonerano dallo sviluppo dell'elaborato progettuale finale.
La prova orale consta di un colloquio, che include anche la discussione di uno dei 3 progetti sviluppati durante il corso o dell'elaborato progettuale finale.

b) Modalità di valutazione:

Per il modulo di Algoritmi e Strutture Dati, le n.2 prove scritte pesano ciascuna il 30% sul giudizio finale (per un totale del 60%). La prova orale pesa per il restante 40% sul giudizio finale (con un peso del 30% per la valutazione e discussione degli esercizi e del 10% per l'interrogazione sulla parte del programma non coperta dalle prove e dagli esercizi).

Per il modulo di Machine Learning, la discussione dell'elaborato progettuale finale (o del progetto sviluppato durante il corso) pesa per il 10% sul giudizio finale, mentre la prova orale per il 90%.

Il voto finale dell'insegnamento sarà ponderato in base ai CFU di ciascun modulo e quindi così composto:
- Modulo di Algoritmi e Strutture Dati, 6 CFU, 50%
- Modulo di Machine Learning, 6 CFU, 50%

SCHEDA DELL'INSEGNAMENTO (SI)

"ARCHITETTURA E PROGETTO DEI CALCOLATORI"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: NICOLA MAZZOCCA
TELEFONO:
EMAIL: NICOLA.MAZZOCCA@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): I
SEMESTRE (I, II): II
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

EVENTUALI PREREQUISITI

Conoscenza dell'architettura di un calcolatore, dei sistemi operativi e delle reti di comunicazione.

OBIETTIVI FORMATIVI

Il corso si pone l'obiettivo di fornire gli elementi metodologici, progettuali e tecnologici per la realizzazione di sistemi di elaborazione con riferimento alle architetture pipelined, multi-computer, multi-processore, multi-core e multi-threading. Il corso affronta inoltre il funzionamento e dimensionamento dei sistemi di memoria gerarchici, il progetto e la programmazione delle unità di I/O (parallele, seriali, DMA e PIC) con i relativi protocolli di comunicazione, e le problematiche di implementazione dei meccanismi di base per la virtualizzazione delle risorse hardware (meccanismi di gestione dei processi, macchine virtuali e hypervisor). Il corso presenta, infine, le principali tecniche per la realizzazione di sistemi pervasivi, autonomici, IoT e di edge computing, nonché le architetture cloud.
La parte applicativa del corso è dedicata al progetto di driver di I/O e allo sviluppo di sistemi operanti in ambito industriale. Le attività vengono svolte con riferimento ad applicazioni sviluppate e valutate sperimentalmente mediante architetture che prevedono l'impiego di nodi di elaborazione dotati di processori RISC e di diversi dispositivi di I/O opportunamente configurabili.
Con riferimento agli aspetti tecnologici, sono illustrate le architetture di sistemi commerciali per l'implementazione di applicazioni industriali basate su System on Chip o su nodi di elaborazione ottenuti per integrazione di componenti configurabili.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e comprendere le problematiche relative al progetto di sistemi di elaborazione, con particolare riferimento alla gestione degli hazard derivanti dall'impiego di tecniche di parallelismo interno ed esterno per l'aumento delle prestazioni, al dimensionamento delle memorie, e all'orchestrazione di diversi sottosistemi operanti in concorrenza fra loro e comunicanti mediante diverse interfacce di I/O.
Deve inoltre dimostrare di saper individuare, fra i diversi approcci presentati al corso, quelli che meglio si adattano a specifiche applicazioni o condizioni operative.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di progettare e sviluppare il software di base (driver assembly) necessario per consentire la comunicazione fra diversi sottosistemi mediante i dispositivi di I/O presentati al corso, anche in presenza di accessi concorrenti a dati comuni, nonché di scheduler per la gestione della concorrenza. Deve inoltre essere in grado di completare il ciclo di sviluppo di applicazioni di media complessità, che richiedano l'utilizzo di uno o più nodi di elaborazione, di diversi dispositivi di I/O, di sensori/attuatori, sui dispositivi hardware in dotazione.

PROGRAMMA-SYLLABUS

Richiami ed approfondimenti sui sistemi di elaborazione: Sistemi general purpose ed embedded. Processori RISC e CISC. Unità di controllo cablata e microprogrammata. Meccanismi di gestione delle interruzioni. Introduzione al parallelismo e al pipelining. Richiami sul processore Motorola 68000. Il processore MIPS: modello di programmazione e pipeline. Il processore ARM. Architetture e applicazioni dei DSP.
Pipelining e hazard: Tecniche di gestione dei conflitti sui dati, dei salti e delle interruzioni in una architettura pipelined. Architetture superscalari.
Sistemi multiprocessore e multicomputer: Architetture parallele, speed up ed efficienza. Algoritmi per la coerenza della memoria.

Periferiche di I/O e driver: Architettura e funzionamento di periferiche parallele, seriali, DMA e PIC, e sviluppo di driver per la loro programmazione.
La gerarchia della memoria: Architettura, indirizzamento e dimensionamento di una cache. Memoria virtuale. Memorie statiche e dinamiche.
Bus e reti di interconnessione: I bus di sistema. Protocolli di comunicazione. Reti di interconnessione: switch multistadio.
Progetto e sviluppo di sistemi basati su microcontrollori: Principi di progetto di sistemi di elaborazione per applicazioni industriali basati su microcontrollori. Architetture e impiego dei System on a Chip (SoC). Dispostivi commerciali e industriali programmabili. Ambienti di progettazione, di simulazione e analisi di sistemi di elaborazione.
Virtualizzazione e cloud computing. Tecniche di virtualizzazione e hypervisor. Introduzione ai sistemi cloud: modelli di servizio e applicazioni.
IoT/Edge computing. Architetture e applicazioni di sistemi IoT e di edge computing. Sviluppo di sistemi edge di tipo commerciale e integrazione di reti di sensori.

MATERIALE DIDATTICO

Libro di testo: Conte, Mazzeo, Mazzocca, Prinetto. Architettura dei calcolatori. Edizioni CittàStudi. 2014. ISBN: 9788825173642.
Dispense e presentazioni fornite dai docenti relative ad argomenti teorici e applicativi trattati al corso.
Manuali e datasheet dei dispositivi utilizzati per l'implementazione delle applicazioni.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso prevede circa il 70% di lezioni frontali in cui vengono affrontati gli argomenti teorici, mentre il restante 30% è riservato a lezioni pratiche ed esercitazioni riguardanti lo sviluppo di driver di I/O e l'utilizzo degli ambienti di sviluppo.
La parte applicativa del corso si avvale di strumenti di sviluppo professionali di cui è disponibile una licenza ad uso gratuito e di board di sviluppo (dotate di un microcontrollore ARM e di diversi dispositivi di I/O) che vengono distribuiti agli studenti per l'implementazione dei propri progetti.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità d'esame:

L'esame si articola in prova	
scritta e orale	x
solo scritta	
solo orale	
discussione di elaborato progettuale	x
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	x
	Esercizi numerici	

La verifica dell'apprendimento prevede una prova scritta consistente in esercizi di progetto di sistemi basati su dispositivi di I/O e una prova orale orientata alla verifica della comprensione dei concetti teorici del corso e alla discussione degli esercizi implementati su board di sviluppo.

SCHEDA DELL'INSEGNAMENTO (SI)

"NETWORKS & CLOUD INFRASTRUCTURES"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: GIORGIO VENTRE
TELEFONO:
EMAIL: GIORGIO.VENTRE@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II): II
SEMESTRE (I, II): I
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dal Regolamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Scopo del corso è fornire le necessarie competenze avanzate sia metodologiche che operative sulla progettazione e gestione di reti di calcolatori complesse e su architetture di elaborazione distribuite su Internet quali Sistemi Cloud ed architetture CDN. Gli obiettivi formativi sono quelli di fornire: i concetti di trasmissione con garanzia di qualità di servizio; le architetture interne dei sistemi di rete; le principali architetture di interconnessione sia per reti metro e wide-area, sia all'interno di data-center e sistemi HPC; le tecnologie e le metodologie per la traffic engineering; i problemi relativi all'internetworking attraverso strutture complesse e multidomain; le tecniche avanzate per il routing sia intra-domain che inter-domain; la creazione di servizi ed applicazioni di rete basati sull'approccio Software Defined Networks; le strategie di network design con caratteristiche di affidabilità; la definizione ed il controllo di service level agreement; i concetti base per il Cloud Computing; i modelli di servizi Cloud (IaaS, PaaS, SaaS). I modelli di attivazione di servizi Cloud (pubblico, privato, ibrido, community); la scalabilità dei servizi Cloud; Network Function Virtualization.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere i concetti relativi alla qualità di servizio in termini di meccanismi di controllo e di gestione. Deve essere in grado di capire la differenza delle varie soluzioni architetturali per la realizzazione di infrastrutture di rete nei vari scenari trattati nel corso e deve avere competenze nei meccanismi di definizione delle metriche e delle politiche di routing. Deve avere conoscenza delle tecniche di Traffic Engineering anche relativamente alla definizione di SLA. Deve essere in grado di comprendere le differenti soluzioni per attivare e adottare servizi basati su architetture Cloud.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di saper applicare le tecniche apprese per la soluzione di problemi avanzati di configurazione di reti e di servizi cloud e di definizione di strategie di traffic engineering.

PROGRAMMA-SYLLABUS

Part I –Network System Architectures

- Switched networks. Switching systems. Input-queued crossbar switches and Head-of-Line blocking. Virtual output queues. Multi-stage interconnection networks. Clos theorem. Spanning Tree Protocol (STP). TRILL.
- Virtual networking. Ethernet VLANs. Virtual network interfaces: TUN/TAP, MacVLAN, and MacVTap. Virtual Ethernet Port Aggregator (VEPA). SR-IOV NICs.
- Software switches: Linux Bridge, Open vSwitch.
- Router architectures and operating systems. Software routers: XORP, Quagga, FRRouting.
- Bare metal, whitebox network switches.
- Data Plane and Control Plane issues. SDN. SDN southbound protocols: OpenFlow. SDN controllers.
- Quality of Service concepts. QoS control and management in packet switched networks.

Part II – Network Infrastructures

- Intra-domain routing. OSPF. IS-IS. ECMP. ECMP in OSPF.
- Inter-domain routing. BGP. SD-WAN. SD-WAN solutions.
- IP networks and virtual circuits. MPLS & switched networks. Traffic Engineering in MPLS networks. RSVP-TE. Segment Routing. Path Computation Element (PCE).
- Evolution of access and carrier networks. Optical networking. SONET/SDH. DWDM. PON and GPON.
- Carrier Ethernet. Metro Ethernet. Ethernet service over MPLS. Ethernet Virtual Private LAN Service (EVPLS).
- Content Delivery Networks.
- Resiliency of Networked Infrastructures. Principles of Fault Tolerant Design.

Part III – Cloud Infrastructures

- Cloud Computing: foundational concepts. Cloud service models (IaaS, PaaS, SaaS). Cloud deployment models (public, private, hybrid, community).
- Elastic computing. Horizontal vs vertical scalability in the cloud. Cloud resources and identity.
- Service Level Agreements. Billing models for cloud services.
- Network Function Virtualization (NFV). The NFV placement problem and solution methods.

MATERIALE DIDATTICO

- Larry Peterson & Bruce Davie, Computer Networks - A system approach. Sixth Edition, Morgan Kauffman, 2021 – ISBN: 978-0128182000
- Lucidi delle lezioni

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso consiste di: a) lezioni frontali per circa l'80% delle ore totali; b) esercitazioni pratiche per il rimanente 20%.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità d'esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

L'esame consiste di una prova orale e della discussione di un elaborato progettuale.

SCHEDA DELL'INSEGNAMENTO (SI)

"IMPIANTI DI ELABORAZIONE"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: DOMENICO COTRONEO
TELEFONO: 0817683824
EMAIL: DOMENICO.COTRONEO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): I
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno.

EVENTUALI PREREQUISITI

Conoscenza dell'architettura di un calcolatore, dei sistemi operativi e delle reti di comunicazione.

OBIETTIVI FORMATIVI

L'obiettivo del corso di Impianti di Elaborazione è quello di fornire gli elementi metodologici per l'analisi e la valutazione quantitativa dei moderni sistemi di elaborazione. Particolare attenzione sarà data sia alle caratteristiche prestazionali sia a quelle di sicurezza e di affidabilità.

Al fine di raggiungere i suoi obiettivi in maniera efficace, il corso presenta, da una prospettiva squisitamente ingegneristica, i seguenti argomenti:

- le metodologie di analisi delle prestazioni e della sicurezza dei sistemi, come ad esempio la statistica inferenziale, l'*experimental design*, la simulazione e i modelli analitici con e senza stato;
- le metodologie e le tecniche di misura diretta di sistemi di larga scala, come ad esempio i data center, con particolare enfasi alle metodologie e le tecniche di *field data Analysis*;
- le metodologie e gli standard per la progettazione e realizzazione dei sistemi *safety critical*.

La parte applicativa del corso è dedicata alla valutazione delle prestazioni e dell'affidabilità di alcuni sistemi, eseguita durante le lezioni; e all'analisi dei dati finalizzata alla valutazione dell'affidabilità e alla sicurezza di sistemi reali di larga scala (data center o sistemi per l'High Peformance Computing).

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e comprendere le problematiche per l'analisi delle prestazioni e dell'affidabilità dei sistemi, con particolare riferimento alle tecniche statistiche di valutazione e di *experimental design*, e alle tecniche di misure diretta. Lo studente deve inoltre dimostrare di conoscere i fondamenti dell'ingegneria dei sistemi *safety-critical*: l'analisi degli *hazard*; l'identificazione e la gestione dei rischi; le tecniche per l'identificazione dei fallimenti.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di valutare le prestazioni e l'affidabilità di un sistema di elaborazione, di comparare le prestazioni e l'affidabilità di due o più sistemi, di conoscere le principale tecniche statistiche per l'analisi dei dati, provenienti da sistemi di larga scala (data center o sistemi per l'High Peformance Computing)

PROGRAMMA-SYLLABUS

CONTENUTI

Fondamenti di statistica Inferenziale applicata all'ingegneria dei sistemi di elaborazione. Distribuzioni campionarie. Intervalli di confidenza. Test di ipotesi. Modelli di regressione. La regressione lineare. Design Of Experiments. Simple Design. Full Fractorial Design. Fractional Factorial Design. ANOVA. Brevi cenni sui tool per elaborazione statistica dei dati: Matlab (statistical toolbox) e JMP.

Metodologie e Tecniche per l'Analisi Sperimentale. . Capacity Planning e Capacity Management. Tipologie di Workload. Application Benchmarks. Criteri per la selezione del workload. Tecniche di caratterizzazione del Workload: Averaging, Single Parameter Histograms, Multi-parameter Histograms Principal Component Analysis, Markov Models e Clustering gerarchico. Capacity Test .

Analisi delle prestazioni di un impianto di elaborazione. Modelli per la misura delle performance e modelli per la misura di attributi di affidabilità. Modelli Analitici e Modelli simulativi. Modelli markoviani a tempo discreto. Modelli markoviani a tempo continuo. Teoria delle code.. Algoritmo di Mean Value Analysis. Esercitazione su alcuni casi di studio.

Analisi dell'affidabilità di un impianto di elaborazione. Sistemi riparabili e non riparabili. Definizione di Dependability: Availability, Reliability, Safety. Misure di Dependability. Fault, Error e Failure. Tecniche di Fault avoidance e tecniche di Fault Tolerance. Duplication. N-Modular Redundancy. Hardware redundancy. Cenni sulle strategie di software Fault Tolerance (N-Version e Recovery Block). Cenni sulla sicurezza dei sistemi cyber physical.

Modelli per la valutazione dell'affidabilità. I modelli combinatoriali: Reliability Block Diagrams e Fault Trees. Modellazione e valutazione di un TMR. Esempi di modellazione di sistemi reali (es. reti di comunicazione bancarie e velivoli).
Metodi e tecniche Di Field Failure Data Analysis: FFDA nel contesto del data mining. Data Collection. Data Filtering. *Data correlation and Coalescence*. Failure analysis di sistemi di larga scala. Casi di studio reali. Supporto all'analisi dei dati con linguaggio Python
Sistemi safety critical. Definizione di Hazard e di Rischio. Hazard Analysis. La FMEA. Risk analysis e management. Cenni sugli Standard per la Safety: IEC61508, DO178B, CENELC EN 50128.
Cenni sulla gestione di un impianto di elaborazione: La scrittura di un capitolato tecnico. Il collaudo di un impianto. Aspetti caratterizzanti la gestione di un impianto di elaborazione. *Business continuity plan* e Tecniche di *Disaster Ricovery*. Tecniche *di Backup e di Restore*

MATERIALE DIDATTICO

Libri di testo:

- Raj Jain, The art of Computer systems Performance Analysis, Wiley
- Douglas C. Montgomery, George C. Runger - Applied Statistics and Probability for Engineers-Wiley
- KishorTrivedi, Andrea Bobbio Reliability and Availability Engineering, Cambridge University Press

Dispense didattiche (consultabili dal sito del corso)
Dispense e presentazioni fornite dal docente relative ad argomenti teorici e applicativi trattati al corso.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso prevede circa il 70% di lezioni frontali in cui vengono affrontati gli argomenti teorici, mentre il restante 30% è riservato a lezioni pratiche ed esercitazioni riguardanti l'applicazione delle tecniche acquisiti ai sistemi reali. La parte applicativa del corso si avvale di tool per elaborazione statistica dei dati , come Matlab, R e JMP.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	x
discussione di elaborato progettuale	x
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

SCHEDA DELL'INSEGNAMENTO (SI)

"TRASMISSIONE DEI SEGNALI DIGITALI"

SSD ING-INF/03

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2025-2026

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: DARSENA DONATELLA
TELEFONO: 081-7683156
EMAIL: darsena@unina.it

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): I
SEMESTRE (I, II): I
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Obiettivo dell'insegnamento è quello di introdurre il tema della trasmissione dell'informazione tra due punti. L'insegnamento si propone di fornire agli studenti nozioni specialistiche sulla trasmissione digitale dell'informazione, che riveste un ruolo fondamentale in ogni sistema di elaborazione dell'informazione, fornendo metodologie per l'analisi e criteri di dimensionamento dei principali parametri che descrivono sinteticamente la qualità del trasferimento informativo

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve conoscere le principali problematiche relative alla descrizione sintetica del canale a forme d'onda, alla conversione di frequenza, ed alla segnalazione e ricezione su un canale a forme d'onda. Deve dimostrare di comprendere i legami tra l'elaborazione del segnale ricevuto e la probabilità di errore che caratterizza il trasferimento informativo, ricordare il suo legame con il livello energetico del rumore di fondo che si sovrappone al segnale ricevuto, illustrare il legame che sussiste tra il tasso binario a cui avviene il trasferimento informativo e la banda disponibile sul canale di forme d'onda.

Capacità di applicare conoscenza e comprensione

Lo studente deve saper risolvere problemi concernenti la valutazione della probabilità di errore associata ad uno specifico meccanismo di ricetrasmissione dell'informazione, anche ricorrendo a strumenti dedicati alla elaborazione al calcolatore di segnali informativi, e di sviluppare algoritmi di elaborazione manipolando schemi tipici introdotti durante il corso.

PROGRAMMA-SYLLABUS

Rappresentazione complessa in banda base dei segnali. Descrizione spettrale dei processi aleatori. Modulazione digitale dei segnali. Il test di ipotesi. Spazio dei segnali. Rivelazione ottima su canale AWGN. Principi fondamentali di teoria della stima. Stima dei parametri con applicazione al problema della sincronizzazione. Comunicazione digitale su canali selettivi in frequenza. Rivelazione a massima verosimiglianza di sequenze di simboli. Algoritmo di Viterbi. Principi di base delle tecniche di equalizzazione lineare. Elementi fondamentali della teoria dell'informazione e della codifica di canale. Laboratorio Matlab (SDR/Matlab).

MATERIALE DIDATTICO

Upamanyu Madhow, Introduction to communication systems, Cambridge, 2014.
Appunti delle lezioni.
Upamanyu Madhow, Fundamentals of Digital Communication, Cambridge, 2008.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso è costituito da lezioni per 72 ore complessive, dedicate alla trattazione degli argomenti teorici e ad attività di laboratorio (SDR/Matlab).

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	
altro	

SCHEDA DELL'INSEGNAMENTO (SI)

"TEORIA DELL'INFORMAZIONE"

SSD ING-INF/03

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: AUGUSTO AUBRY
TELEFONO: 081-76-83148
EMAIL: AUGUSTO.AUBRY@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE): TEAMS: "TEORIA DELL'INFORMAZIONE – ANNO ACCADEMICO 2021-2022"
ANNO DI CORSO (I, II, III): I/II LAUREA MAGISTRALE
SEMESTRE (I, II): I
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dal Regolamento del CdS)

In quanto corso di laurea magistrale, l'insegnamento èrivolto a studenti che abbiano conseguito una qualsiasi laurea triennale nella classe delle Ingegnerie dell'Informazione o in Informatica.

EVENTUALI PREREQUISITI

Nessuno.

OBIETTIVI FORMATIVI

Il corso ha come obiettivo fondamentale la definizione - da un punto di vista quantitativo - dell'informazione, in modo da consentire allo studente di comprendere le scelte progettuali alla base del dimensionamento dei sistemi di immagazzinamento, elaborazione e trasmissione dell'informazione. Lo studente acquisirà quindi familiarità con il concetto di "rimozione della ridondanza" (alias codifica di sorgente o compressione senza perdite), di compressione con perdite (comunemente detta quantizzazione, sia scalare che vettoriale), nonchè con concetti quali codifica di canale e capacità di canale. Obiettivo non secondario è quello di illustrare alcuni codici - soprattutto di sorgente - di particolare importanza applicativa, quali la codifica aritmetica e la codifica universale. Parallelamente, il corso sviluppa alcuni concetti basilari di statistica inferenziale e di ottimizzazione.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente sarà in grado di comprendere le ragioni di innumerevoli scelte progettuali compiute nell'ambito dei sistemi di trasmissione, immagazzinamento ed elaborazione dell'informazione, i principi fondamentali che regolano il progetto di codici sia di sorgente che di canale, nonché alcune delle metodologie fondamentali per eseguire l'analisi statistica dei dati.

Capacità di applicare conoscenza e comprensione

Lo studente imparerà a considerare l'informazione come una qualsiasi altra grandezza ingegneristica, ed acquisirà gli strumenti per il dimensionamento di sistemi di memorizzazione, trasmissione ed elaborazione dell'informazione,nonché dell'impiego di alcune metodologie fondamentali della statistica inferenziale.

PROGRAMMA-SYLLABUS

Generalità. Informazione e sua misura. Entropia, entropia congiunta, entropia condizionale. Mutua informazione e sue proprietà. Divergenza e sue interpretazioni. Estensione alle variabili continue.
Le sorgenti informazionali. Modello di una generica sorgente informazionale. Regola della catena per l'entropia e tasso entropico. Concetto di tipicità. Comprimibilità delle sorgenti.
Compressione senza perdite. Codici di sorgente e loro classificazione. Disuguaglianza di Kraft e sue conseguenze. Limiti fondamentali sulla lunghezza media dei codici di sorgente. Codici di Shannon, Huffman, e Shannon-Fano-Elias. Codifica aritmetica. Codici di Lempel-Ziv.
Compressione con perdite. Il concetto di distorsione. Principali misure di distorsione: Hamming e distorsione quadratica. Trade-off tra tasso (rate) e distorsione. La funzione rate-distortion e sue interpretazioni. Calcolo della funzione R(D) per alcuni casi notevoli. Estensione del concetto di tipicalità. Rate-Distortion Theorem.
Capacità di canale.
a) Canali generici. Modello generale di canale di comunicazione e concetto di capacità. Calcolo della capacità di alcuni canali notevoli. Teorema della capacità di canale: proposizione diretta e proposizione inversa. Cenni alla codifica algebrica. Il teorema di separabilità.
b) Canali Gaussiani. Il canale gaussiano additivo: calcolo della capacità, prova della proposizione diretta e inversa. Canali Gaussiani in parallelo e waterfilling. Canali con rumore gaussiano correlato. Canali gaussiani con feedback.
Elementi di Statistica Inferenziale. Significato di inferenza. Caratterizzazione a priori e a posteriori. Elementi di statistica Bayesiana e non-Bayesiana. Limiti di Rao-Cramer. Stima LMMSE. Generalità sull'algoritmo del gradiente.
Elementi di tecniche di Ottimizzazione. Definizione di problema di ottimizzazione vincolata.
Total least squares..

MATERIALE DIDATTICO

T. M. Cover, J. A. Thomas, *Elements of Information Theory*, 2nd Edition, J. Wiley &Sons, 2006
(vedi note per le parti sviluppate nel corso).

Slides e appunti integrativi preparate dal docente, disponibili sul canale MSTeams dedicato.
Copia delle "lavagne" contenenti le derivazioni eseguite in classe nel corso delle lezioni erogate telematicamente (se previste).

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso prevede circa settanta ore di lezioni frontali, di cui 2/3 del tempo saranno dedicati alla parte teorica e metodologica e 1/3 ad applicazioni ed esercitazioni numeriche.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	
altro	

b) Modalità di valutazione:

L'esame consisterà in un colloquio orale.

SCHEDA DELL'INSEGNAMENTO (SI)

"ARCHITETTURA DEI SISTEMI INTEGRATI"

SSD ING-INF/01

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ANTONIO G. M. STROLLO
TELEFONO: 081-7683125
EMAIL: mailto:dadecaro@unina.itANTONIOGIUSEPPEMARIA.STROLLO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

ANNO DI CORSO (I, II, III): I
SEMESTRE (I, II): I
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno.

EVENTUALI PREREQUISITI

Conoscenza di base dei circuiti digitali, delle principali caratteristiche dei dispositivi MOS e delle logiche CMOS.

OBIETTIVI FORMATIVI

Nell'ambito del corso viene studiato il flusso di progetto dei circuiti integrati digitali, a partire dalla descrizione mediante linguaggi per la descrizione dell'hardware fino all'implementazione fisica. L 'insegnamento si pone l'obiettivo di fornire agli studenti le metodologie e le conoscenze utili a disegnare i moderni microcircuiti ad alta scala di integrazione, valutarne le caratteristiche, ottimizzarne le prestazioni e definirne le procedure di verifica.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il percorso formativo intende fornire agli studenti le conoscenze e gli strumenti metodologici necessari a completare il flusso di sviluppo di un sistema digitale integrato. In particolare, lo studente sarà edotto sugli standard utilizzati per realizzare descrizioni sintetizzabili utilizzando linguaggi per la descrizione dell'hardware, sulle tecniche di valutazione ed ottimizzazione dei ritardi e della dissipazione di potenza, sulle varie architetture per implementazione di circuiti aritmetici e sulle principali metodologie di testing. Tali strumenti consentiranno agli studenti di comprendere le principali relazioni che sussistono tra l'implementazione fisica dei sistemi integrati e le loro caratteristiche elettriche.

Capacità di applicare conoscenza e comprensione

Al termine del processo di apprendimento lo studente è in grado di progettare ed analizzare a livello architetturale, circuitale e fisico circuiti e sistemi digitali ad alta scala di integrazione. Egli disporrà degli strumenti metodologici e operativi necessari a descrivere mediate linguaggi HDL un sistema digitale, ad approntarne il test-bench ed a realizzarne la simulazione. Lo studente sarà inoltre in grado di definire gli opportuni vincoli necessari ad effettuare la fase di sintesi, di utilizzare programmi di sintesi automatica e di valutare i risultati forniti dal sintetizzatore. Il percorso formativo è orientato a trasmettere le capacità e gli strumenti metodologici e operativi necessari ad applicare concretamente le varie tecniche volte ad ottimizzare le prestazioni dei sistemi digitali, in termini di velocità, area occupata e consumo energetico.

Autonomia di giudizio, Abilità comunicative, Capacità di apprendimento

Al termine del corso lo studente deve essere in grado di sapere scegliere in maniera autonoma le possibili metodologie e tecniche da utilizzare nelle varie fasi del ciclo di progettazione di un sistema digitale ad alta scala di integrazione; dovrà inoltre avere la capacità di valutare i risultati derivanti dall'applicazione delle varie tecniche di ottimizzazione ed essere in grado di confrontare le prestazioni di diverse architetture.

Gli studenti saranno in grado di approfondire autonomamente argomenti trattati. La metodologia di verifica ed il confronto con il docente tendono inoltre a sviluppare le abilità comunicative degli studenti che devono dimostrare di saper impostare una relazione scientifica utilizzando terminologia e linguaggio appropriato.

PROGRAMMA-SYLLABUS

Classificazione dei sistemi integrati: full-custom, basati su celle standard e programmabili. Metodologie di progetto di sistemi integrati. Tecniche automatiche di sintesi e di piazzamento e collegamento di celle standard.

Caratteristiche dei transistori MOS di ultima generazione. Tecniche di simulazione switch-level. Valutazione semplificata dei ritardi delle porte logiche.

Analisi statica dei ritardi. Grafi dei ritardi. Caratterizzazione dei ritardi delle celle standard. Livelli di interconnessione e parametri parassiti. Valutazione dei ritardi introdotti dalle interconnessioni mediante la tecnica di Elmore. Ripetitori. Effetti dello scaling tecnologico sui ritardi delle interconnessioni. Cross-talk. Distribuzione delle linee di alimentazione.

Progetto e temporizzazione di sistemi sequenziali. Tempi caratteristici dei registri. Registri avanzati. Tecniche di pipelining. Effetti delle non idealità del clock (skew, jitter) sulla temporizzazione dei sistemi sequenziali. Generazione e distribuzione del clock. Anelli ad aggancio di fase (PLL) e ad aggancio di ritardo (DLL).

Valutazione della dissipazione di potenza nei sistemi VLSI. Fonti di dissipazione di potenza statica e dinamica. Tecniche per la riduzione della dissipazione di potenza a livello tecnologico, circuitale ed architetturale.

Il linguaggio VHDL per la descrizione e la sintesi di sistemi integrati. Statements sequenziali e concorrenti. La simulazione event-driven. Librerie standard per la sintesi di sistemi digitali. Descrizione e sintesi di circuiti aritmetici. Test-bench. Operazioni su file di testo.
Testing dei sistemi integrati CMOS. Modelli di guasto. Algoritmi per il calcolo dei vettori di test. Tecniche di self-test.
Circuiti aritmetici. Addizionatori a selezione del ritardo, carry-skip, parallel-prefix. Addizionatori multi-operando. Moltiplicatori paralleli. Moltiplicatori veloci (Wallace, Dadda).

MATERIALE DIDATTICO

- Weste, Harris, "CMOS VLSI Design – circuit and systems perspective", 4th edition, Pearson – Addison Wesley, 2011
- Appunti delle lezioni
- Testi delle esercitazioni

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso prevede lezioni frontali, esercitazioni e, compatibilmente con gli aspetti organizzativi, esercitazioni di laboratorio.
Per lo svolgimento delle esercitazioni gli studenti adottano programmi di sviluppo di sistemi integrati della Cadence, resi disponibili grazie alle "Cadence Low-Cost Classroom Teaching Licenses" acquisite mediante Europractice. Gli studenti inoltre utilizzano un simulatore vhdl (ghdl) ed un visualizzatore di forme d'onda (gtkwave) open-source.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	✓
discussione di elaborato progettuale	
altro (discussione esercitazioni)	✓

"ALGORITMI DI OTTIMIZZAZIONE COMBINATORIA E SU RETE"

SSD MAT/09

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: MAURIZIO BOCCIA
TELEFONO: 081 7683247
EMAIL: MAURIZIO.BOCCIA@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

ANNO DI CORSO: I
SEMESTRE: II
CFU: 9

INSEGNAMENTI PROPEDEUTICI

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Il corso ha l'obiettivo di fornire agli studenti gli strumenti metodologici per analizzare e risolvere problemi di programmazione matematica con particolare riferimento ai problemi di ottimizzazione combinatoria e di ottimizzazione su rete. Vengono presentate metodologie risolutive sia per taluni problemi "di base" dell'ottimizzazione su reti, sia per problemi "difficili" di ottimizzazione combinatoria e su reti. Al termine del corso lo studente avrà acquisito la capacità di formulare un modello astratto di un problema di ottimizzazione combinatoria, la capacità di individuare le strutture presenti su cui articolare un approccio risolutivo di tipo euristico, oppure, di applicare un algoritmo ad hoc per la sua soluzione esatta.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il percorso formativo ha l'obiettivo di fornire agli studenti le metodologie di ottimizzazione combinatoria e di ottimizzazione su rete necessarie per la modellazione e la risoluzione esatta e/o euristica di problemi decisione che possono presentarsi in ambito ingegneristico e industriale. Al termine del corso, lo studente dovrà dimostrare di conoscere e di saper utilizzare gli strumenti necessari a formulare un problema di ottimizzazione combinatoria e di ottimizzazione su reti. Deve inoltre essere in grado di decidere, in relazione alle caratteristiche e alla complessità del problema, se la risoluzione del problema possa avvenire mediante l'utilizzo dei pacchetti software e delle librerie di ottimizzazione più diffusi, oppure occorra sviluppare un algoritmo ad hoc per la sua soluzione. Deve quindi dimostrare di conoscere le principali metodologie meta-euristiche per la risoluzione di problemi di ottimizzazione combinatoria e saper scegliere la metodologia più adatta al problema da affrontare.

Capacità di applicare conoscenza e comprensione

Il percorso formativo è orientato a trasmettere agli studenti gli strumenti metodologici e operativi necessari alla formulazione e alla soluzione di problemi di ottimizzazione combinatoria e di problemi di ottimizzazione su reti. In particolare, lo studente deve dimostrare di saper formulare un problema di ottimizzazione. Deve inoltre dimostrare di saperlo risolvere, quando possibile, in maniera esatta utilizzando una delle principali librerie di ottimizzazione, oppure in maniera approssimata utilizzando un approccio di tipo euristico. Infine, deve essere in grado di implementare e sperimentare i diversi approcci euristici e meta-euristici allo scopo di decidere l'approccio più adatto al particolare problema.

PROGRAMMA-SYLLABUS

I Modelli della Ricerca Operativa

- L'approccio modellistico
- Modelli di Ottimizzazione

Programmazione Lineare continua e intera

- Introduzione alla Programmazione Lineare
- Esempi di modelli di programmazione lineare
- Rappresentazione grafica di un problema di P.L.
- Formulazioni.
- Il metodo Branch and Bound
- Il metodo dei piano di taglio

- Il metodo Branch and Cut

Software per la Programmazione Matematica
- Utilizzo della libreria Gurobi per la soluzione di problemi di PL e PLI.
- Esempi d'utilizzo della libreria: pianificazione della produzione, gestione delle scorte, problemi su reti, problemi di localizzazione, problemi di distribuzione.

Euristiche per la soluzione di problemi di ottimizzazione combinatoria
- L'ottimizzazione combinatoria
- Euristiche di tipo greedy
- Ricerca locale e tabù search
- Algoritmi genetici

Elementi di teoria dei grafi
- Forme di rappresentazione di un grafo
- Algoritmi di visita di un grafo

Il problema di instradamento e flussi su rete
- Classificazione dei problemi e degli algoritmi di minimo percorso
- Il modello del minimo percorso
- Algoritmi per il calcolo dei minimi percorsi
- Problemi di flusso Single e Multi-Commodity
- Politiche di instradamento e bilanciamento di una rete

Problemi di design, localizzazione e clustering
- Problemi di localizzazione su nodi e su archi
- Euristica costruttive e migliorative
- Il problema di clustering
- Algoritmo k-means per il problema di clustering

Il problema del Commesso Viaggiatore (TSP)
- Complessità del problema
- Un algoritmo di row generation
- Euristiche greedy e di ricerca locale
- Il problema di orienteering come variante del TSP

Il problema di Vehicle Routing
- Definizione del problema e delle sue varianti
- Euristiche greedy e di ricerca locale

MATERIALE DIDATTICO

- M. Caramia, S. Giordani, F. Guerriero, R. Musmanno, D. Pacciarelli, "Ricerca Operativa", Isedi, Italia, 2014.
- H. Paul Williams, Model Building in Mathematical Programming, John Wiley & Sons, Ltd.
- F. S. Hillier, G. J. Lieberman, Ricerca operativa - Fondamenti, 9/ed., McGraw-Hill, 2010.
- A. Sforza, Modelli e Metodi della Ricerca Operativa, 3a ed., ESI, Napoli, 2018.
- G. Bruno, Operations Management. Modelli e metodi per la logistica, ESI – Edizioni Scientifiche Italiane.
- Materiale didattico integrativo fornito durante il corso.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà: lezioni frontali (65%), seminari (5%), esercitazioni di tipo numerico (10%), esercitazioni di utilizzo di librerie di ottimizzazione (25%). Il materiale del corso sarà reso disponibile on-line agli studenti.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	X
solo scritta	
solo orale	
discussione di elaborato progettuale	X
altro	

In caso di prova scritta i quesiti sono	**A risposta multipla**	
	A risposta libera	X
	Esercizi numerici	X

b) Modalità di valutazione:

La prova scritta è volta a verificare le capacità di formulazione di problemi di ottimizzazione e la comprensione degli algoritmi di risoluzione di problemi PL e PLI. Lo studente ha a disposizione 2 ore per la prova scritta. L'esame prevede inoltre lo svolgimento di un elaborato progettuale in cui lo studente deve implementare e sperimentale un algoritmo esatto e/o euristico per la soluzione di un problema di ottimizzazione combinatoria. Il colloquio orale avrà come oggetto sia la discussione dell'elaborato progettuale che l'accertamento dell'acquisizione dei concetti e delle metodologie illustrati durante le lezioni.

L'esito della prova scritta è vincolante ai fine dell'accesso alla prova orale. La prova scritta e quella orale con la descrizione dell'elaborato finale contribuiscono rispettivamente per il 40% e il 60% della valutazione finale. Il superamento della prova scritta non è sufficiente per il superamento dell'esame.

SCHEDA DELL'INSEGNAMENTO (SI)

"CALCOLO SCIENTIFICO PER L'INNOVAZIONE TECNOLOGICA"

SSD MAT/08

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: PROF. LUISA D'AMORE
TELEFONO: 081-675626
EMAIL: LUISA.DAMORE@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE): NO
MODULO (EVENTUALE):UNICO
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III):I
SEMESTRE (I, II) II
CFU:9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

NESSUNO

EVENTUALI PREREQUISITI

NESSUNO

OBIETTIVI FORMATIVI

Le simulazioni numeriche al calcolatore (che Galileo Galilei chiamava analisi matematica e sperimentale della natura) consentono di dominare la crescente complessità dei modelli delle scienze e dell'ingegneria: esplorando soluzioni innovative mediante l'uso intensivo di tecnologie hardware e software emergenti, hanno un impatto sociale decisivo (policy decision-making). La simulazione numerica è il risultato di un processo in continua evoluzione. Simulazioni affidabili, ovvero in grado non solo di interpretare ma anche di prevedere, devono tener conto dell'incertezza. L'analisi dell'incertezza, della sua propagazione e lo studio della sensibilità della soluzione alle incertezze nei dati (Uncertainty Quantification) è un requisito fondante delle simulazioni predittive.
L'insegnamento intende approfondire conoscenze metodologiche della matematica numerica (alla base delle simulazioni numeriche) preliminari e fondamentali per calcolare, interpretare e descriverne le soluzioni. L'analisi critica dei metodi numerici, degli algoritmi risolutivi e la loro messa in opera in ambienti di calcolo come il Matlab (anche Parallelo), consentirà agli studenti di identificare l'approccio migliore per ottenere il risultato atteso.

Gli obiettivi formativi dell'insegnamento sono i seguenti

- Acquisire conoscenze metodologiche specialistiche del *mathematics stack* per le simulazioni e le modellazioni al calcolatore (dalla formulazione del problema alla interpretazione, visualizzazione, verifica e validazione dei risultati;
- Approfondire gli strumenti della matematica numerica (algoritmi e software), selezionandoli opportunamente in ragione della tipologia del problema e allo strumento di calcolo (co-design);
- Padroneggiare le competenze numeriche necessarie per affrontare in un contesto interdisciplinare e multidisciplinare la modellazione e simulazione computazionale di sistemi complessi di interesse dell'Ingegneria Informatica.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il percorso formativo intende fornire agli studenti le conoscenze metodologiche del calcolo scientifico necessarie alla realizzazione di simulazioni scientifiche dei problemi che nascono in realtà produttive molto differenziate. Lo studente deve dimostrare di conoscere gli aspetti caratterizzanti la quantificazione dell'incertezza nelle simulazioni computazionali ovvero comprendere le problematiche relative alle principali fonti di errore che nascono nella fase di risoluzione computazionale, di riconoscere e ridurne la propagazione sulla soluzione numerica attraverso l'analisi critica degli algoritmi numerici e di mettere a punto elementi di software affidabili in grado di stimare l'incertezza sul risultato. A fine corso, lo studente sarà in grado di definire il processo di risoluzione computazionale di alcuni problemi fondamentali della matematica numerica.

Capacità di applicare conoscenza e comprensione

Il percorso formativo è orientato a fornire le capacità operative necessarie a trasformare un qualsiasi modello matematico, come ad esempio un'equazione differenziale alle derivate parziali, un integrale o un sistema di equazioni lineari, in un elemento di software matematico e di utilizzare gli standard qualitativi per la verifica e la validazione dei risultati.
A fine corso, lo studente sarà in grado di selezionare e utilizzare gli strumenti della matematica numerica (metodi, algoritmi e software) acquisiti per risolvere alcuni casi di studio analizzando, interpretando e validando i risultati ottenuti.

PROGRAMMA-SYLLABUS

Articolazione del corso:
Parte 1 - 30 % – concetti fondamentali
Parte 2 – 20 % – approfondimenti
Parte 3 – 50 % – preparazione alla prova finale

Parte 1:

Introduzione ai principi del Calcolo Scientifico – fasi di risoluzione computazionale di un problema, fonti di errore, stabilità di un algoritmo, condizionamento di un problema. Analisi di propagazione dell'errore.

Algebra Lineare Numerica – Metodi diretti, Algoritmo di Eliminazione di Gauss e Fattorizzazione LU per matrici piene o strutturate.

Parte 2:

Algebra Lineare Specialistica - Metodi iterativi. Il metodo del Gradiente Coniugato Precondizionato. I metodi MULTIGRID. Le librerie BLAS (Basic Linear Algebra Subroutines) e PBLAS (Parallel BLAS). Introduzione alle problematiche relative allo sviluppo di algoritmi paralleli per il calcolo matriciale e analisi di prestazioni (speed up, efficienza). Sperimentazione nell'ambiente Parallel Toolbox di Matlab.

Parte 3:

Rappresentazione di dati – Metodi di interpolazione e approssimazione per il trattamento e l'analisi di dati. Le funzioni spline. Interpolazione di curve mediante spline parametrica. Curve di Bezier, Curve B-spline- NURBS. Rappresentazione di oggetti in due e tre dimensioni. Il Polinomio di minimi quadrati di primo e secondo grado. Introduzione ai modelli numerici per il machine learning. Introduzione agli ambienti software per la modellazione di oggetti.

Algoritmo FFT – Dalla Trasformata di Fourier alla Trasformata discreta di Fourier mediante formule di quadratura numerica. Algoritmo FFT di Gentleman e Sande, Algoritmo FFT di Cooley e Tukey. Esempi ed applicazioni in Matlab: filtraggio/denoising/ricostruzione di un segnale e di una immagine. Algoritmo FFT radix-2, radix-r, a radici miste.

Equazioni Differenziali alle Derivate Parziali: Concetti e definizioni preliminari sugli operatori differenziali alle derivate parziali: **c**lassificazione (degli operatori, delle condizioni iniziali e al contorno, dei problemi differenziali alle derivate parziali); Esempi in contesti applicativi. L'approccio numerico alla base della risoluzione di problemi differenziali alle derivate parziali; Introduzione ai metodi di discretizzazione: differenze finite e volumi finiti; Concetto di consistenza, stabilità e convergenza.

MATERIALE DIDATTICO

- A.D'Alessio, Lezioni di Calcolo Numerico e Matlab- Liguori editore, IIed
- L. D'Amore, Spigolature sul metodo dei Gradienti Coniugati, Libreria Universitaria.
- A.Murli, Matematica Numerica, parte prima. Ed. Liguori

Dispense su argomenti specifici disponibili sul sito del docente

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà

a) Lezioni frontali per circa il 30%
b) Esercitazioni per circa il 20%
c) Laboratorio per circa il 20%
d) Software specialistico per circa il 30%

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta o intercorso a metà	x
solo orale	
discussione di elaborato progettuale	x
Altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	x

SCHEDA DELL'INSEGNAMENTO (SI)

"ELABORAZIONE DI SEGNALI MULTIMEDIALI"

SSD ING-INF/03

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: LUISA VERDOLIVA
TELEFONO: 0817683929
EMAIL: LUISA.VERDOLIVA@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE): NO
MODULO (EVENTUALE): -
CANALE (EVENTUALE): -
ANNO DI CORSO (I, II, III): I-II
SEMESTRE (I, II): II
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Obiettivo dell'insegnamento è di fornire la conoscenza dei concetti di base e degli algoritmi per l'elaborazione di immagini digitali e presentare le principali tecniche per la codifica di immagini fisse e sequenze video, con particolare attenzione agli standard più comuni. Oltre a fornire gli strumenti matematici e concettuali per trattare analiticamente questi argomenti, il corso si propone di dare le conoscenze necessarie per sviluppare in Python i principali algoritmi per l'elaborazione delle immagini.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e saper comprendere gli strumenti metodologici per l'analisi e l'elaborazione delle immagini. Tali strumenti consentiranno agli studenti di risolvere problemi più complessi sia nel dominio dello spazio che nel dominio della frequenza.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di saper ragionare sulle problematiche riguardo l'analisi e l'elaborazione di immagini e di saper scegliere la tecnica più adatta per la risoluzione di un problema pratico.

PROGRAMMA-SYLLABUS

Enhancement di immagini. Enhancement nel dominio spaziale. Trasformazioni basilari dell'intensità: operazioni puntuali lineari e non lineari. Equalizzazione dell'istogramma. Bit-plane slicing. Operazioni aritmetiche. Operazioni geometriche. Basi del filtraggio spaziale. Filtri di smoothing e di sharpening. Filtro mediano. Operazioni morfologiche. Enhancement nel dominio frequenziale. Trasformata di Fourier bidimensionale. DFT-2D. Esempi di filtro passa-basso e passa-alto. Filtraggio in frequenza: filtri ideali, filtri di Butterworth e filtri gaussiani.

Rappresentazione del colore. Cenni sul sistema visivo umano. Coni e bastoncelli. Sensibilità relativa dei coni di tipo S, M e L. Teoria tricromatica dei colori. Color matching. Gli spazi di colore (RGB, HSI). Pigmenti: colorazione sottrattiva, sistemi di stampa CMY, CMYK (quadricromia).

Segmentazione. Tecniche edge based. Point detection e line detection. Gradienti di Roberts, Prewitt e Sobel. Thresholding del gradiente. Zero-crossing del Laplaciano. Canny edge detector. Tecniche class-based. Algoritmo K-means.

Compressione di immagini. Codifica di sorgente. Generalità sulla compressione dati. La quantizzazione. Quantizzazione uniforme e non uniforme. Codifica predittiva. Schema del codificatore e decodificatore. Quantizzazione predittiva. Codifica mediante trasformata. Compattazione dell'energia e allocazione ottima delle risorse. Le trasformate lineari. Trasformata di Karhunen-Loéve e sue proprietà. Trasformata coseno discreta. Lo standard JPEG.

Codifica video. Generalità sul segnale video. Compressione spaziale e temporale. Conditional replenishment e compensazione del movimento. Il codificatore ibrido. Cenni allo standard MPEG-1 e MPEG-2. La scalabilità in risoluzione e in frame-rate

Trasformata Wavelet. Localizzazione tempo-frequenza. Wavelet continua (CWT). Mother Wavelet. Analisi multirisoluzione, funzione di scaling. Equazioni MRA. Estensione al caso bidimensionale. Implementazione della trasformata wavelet discreta (DWT). Banco di filtri di analisi e sintesi. Codifica mediante Wavelet. Algoritmo EZW.

Applicazioni. Esempi di applicazioni avanzate per l'elaborazione di immagini: denoising, super-risoluzione, riconoscimento di volti o oggetti, classificazione mediante descrittori locali, segmentazione semantica, riconoscimento di manipolazioni nelle immagini e nei video anche con tecniche basate sull'apprendimento (cenni alle reti neurali convoluzionali).

MATERIALE DIDATTICO

Libri di testo consigliati:

- R.C.Gonzalez, R.E.Woods: “Digital image processing”, 3nd edition, Prentice Hall, 2008.
- A.Bovik: “The essential guide to image processing”, Academic Press, 2009.
- K.Sayood: “Introduction to data compression”, 2nd edition, Morgan Kaufmann, 2000.

Appunti del corso di Elaborazione di Segnali Multimediali: http://wpage.unina.it/verdoliv/esm/

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

La didattica è erogata: a) per circa il 70% con lezioni frontali; b) per circa il 30% con esercitazioni guidate in laboratorio per lo sviluppo di applicazioni software in Python per comprendere al meglio le tecniche studiate.
Gli argomenti delle lezioni frontali e delle esercitazioni sono esposti con l’ausilio di lavagne elettroniche e/o trasparenze dettagliate, messe a disposizione dello studente nel materiale didattico tramite il sito web ufficiale del docente. E’ anche prevista la registrazione delle lezioni.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	X
solo scritta	
solo orale	
discussione di elaborato progettuale	
Altro	

L’esame si articola in una prova scritta e una orale. La prova scritta, che consiste di tre algoritmi da sviluppare in Python al calcolatore, può essere sostituita dallo sviluppo di un progetto pratico in Python su un’applicazione avanzata di elaborazione di immagini. La prova orale consiste in due domande su problemi/algoritmi esposti al corso.

SCHEDA DELL'INSEGNAMENTO (SI)

"INFORMATION SYSTEMS AND BUSINESS INTELLIGENCE"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO: 2025-2026

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: FLORA AMATO
TELEFONO: 0817683835
EMAIL: FLORA.AMATO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): I
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Il corso si pone l'obiettivo di fornire le competenze specialistiche per la progettazione e gestione delle moderne infrastrutture di Business Intelligence (BI) a supporto dei Sistemi Informativi di tipo direzionale e viste sia come strumento al servizio del raggiungimento degli obiettivi aziendali, sia come catalizzatore dell'innovazione organizzativa e strategica. In particolare, verranno analizzate alcune delle suite open-source e commerciali più diffuse per la BI ed approfondite le metodologie e tecniche di Data Mining per l'estrazione di informazioni e conoscenza utili ai processi decisionali. In aggiunta, saranno dettagliate anche le moderne tecnologie ERP/CRM a supporto dei processi di business di tipo operativo, e analizzate come queste siano una delle principali sorgenti dati in un sistema di BI. Vengono infine affrontati i principi metodologici di alcune fasi del ciclo di vita di un Sistema Informativo, con riferimento non solo agli aspetti tecnologici, ma anche a quelli che richiedono attenzione al contesto organizzativo ed economico, nonché le relative problematiche di assessment e benchmarking.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il percorso formativo intende fornire agli studenti le conoscenze specialistiche e tutti gli strumenti metodologici e tecnologici a supporto sia dei Sistemi Informativi Direzionali sia di quelli Operativi, con particolare riferimento alle principali suite software ERP/CRM e di Business Intelligence (BI) sia commerciali sia open-source. In particolare, tali strumenti consentiranno agli studenti, da un lato, di sapere utilizzare quelle che sono le "best practices" più adeguate alla messa a norma e reingegnerizzazione dei processi di business di natura operativa, dall'altro, di sapere implementare in maniera efficiente ed efficace i processi direzionali/decisionali di una data organizzazione.

Capacità di applicare conoscenza e comprensione

Il percorso formativo è orientato a trasmettere le capacità e gli strumenti metodologici ed operativi necessari ad applicare concretamente le conoscenze relative alla progettazione e manutenzione dei Sistemi Informativi Aziendali sia Direzionali sia Operativi. In particolare, lo studente deve dimostrare di essere in grado di reingegnerizzare attraverso l'introduzione delle tecnologie informatiche ERP/CRM quelli che sono i tipici processi di business operativi di un'azienda, e di supportare con i tool di Business Intelligence (BI) quelli che sono invece i processi direzionali/decisionali.

PROGRAMMA-SYLLABUS

1. **I Sistemi Informativi Aziendali**: Introduzione ai Sistemi Informativi Aziendali. Il modello organizzativo, funzionale ed informatico di un Sistema Informativo. Sistemi Informativi Operativi vs Sistemi Informativi Direzionali.
2. **I Processi Gestionali nei Sistemi Informativi**: Tipologie di classificazione dei processi. Identificazione, descrizione e scomposizione dei processi. Cenni alla Modellazione ed al Workflow dei Processi. Prestazioni dei Processi Gestionali. Business Process Reengineering (BPR).
3. **Le Tecnologie Informatiche alla base dei Sistemi Informativi Operativi**: Architettura dei moderni Sistemi Informativi. Sistemi ERP. Sistemi CRM. Architetture SOA. Integrazione di Sistemi Informativi.
4. **Sistemi Informativi Direzionali e Business Intelligence**: Definizione di Business Intelligence (BI). Il processo di BI. Principali tool commerciali e open-source. Tecniche di Data Mining a supporto della BI.
5. **Ciclo di Vita dei Sistemi Informativi**: Pianificazione. Assessment e Benchmarking. Reingegnerizzazione e Studio di Fattibilità. Progettazione, Realizzazione e Manutenzione. Gestione e Conduzione. Cenni al Project Management.
6. **Esempi di Sistemi Informativi**: Sistemi Informativi per la Logistica e la Produzione. Sistemi Informativi per i Trasporti. Sistemi informativi Sanitari. Sistemi Informativi per la Pubblica Amministrazione. Sistemi Informativi Territoriali.

MATERIALE DIDATTICO

Libri di testo consigliati:

- Bracchi, Francalanci, Motta, “Sistemi informativi per l’industria digitale”, Mc Graw Hill 2010.
- Rezzani, “Business intelligence. Progessi, metodi, utilizzo in azienda”, Apogeo, 2012.

Appunti del docente, Lucidi.
Manuali dei vari tools presentati.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà lezioni frontali per circa il 60% delle ore totali, ed in aggiunta esercitazioni al calcolatore, sia assistite sia personali, per approfondire praticamente gli aspetti teorici ed i tool (software specialistico) introdotti, e seminari di approfondimento per le rimanenti ore. Il tutto sarà supportato da materiale didattico multimediale disponibile on-line.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	**X**
discussione di elaborato progettuale	**X**
Altro	

Ogni studente dovrà sviluppare durante il corso 2 Homeworks, in aggiunta all’elaborato progettuale finale, comprendenti l’uso delle tecniche e de tools presentati.

b) Modalità di valutazione:

Durante la discussione dell’elaborato progettuale saranno richiesti approfondimenti teorici relativi a parti dell’elaborato stesso, ed il tutto concorrerà alla valutazione finale.

SCHEDA DELL'INSEGNAMENTO (SI)

"AI SYSTEMS ENGINEERING"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ROBERTO PIETRANTUONO
TELEFONO: 0817683880
EMAIL: ROBERTO.PIETRANTUONO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): I
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Algoritmi, Strutture Dati e Machine Learning

EVENTUALI PREREQUISITI

Ingegneria del Software

OBIETTIVI FORMATIVI

Il corso intende fornire le metodologie e gli strumenti necessari per lo sviluppo e l'evoluzione dei sistemi basati su AI, con riferimento all'intero ciclo di vita: progettazione, sviluppo, verifica e validazione, evoluzione/manutenzione, operatività. Il corso illustra le peculiarità dei sistemi basati su AI rispetto ai sistemi tradizionali (quali ad esempio il ruolo fondamentale dei dati), come queste peculiarità impattano sulle varie fasi del ciclo di vita del sistema, e quali processi, modelli, linguaggi, tecniche, tecnologie e strumenti sono richiesti per l'ingegnerizzazione di sistemi di alta qualità.
I contenuti saranno esposti con riferimento ad ambiti applicativi di crescente diffusione, quali ad esempio sistemi per la guida autonoma di autovetture, velivoli senza conducente (*unmanned aerial systems*), (federazioni di) *bot* usati nel *customer service*, nei *social media*, nella realtà virtuale/metaverso. E' previsto l'utilizzo di diversi strumenti software a supporto delle varie fasi del ciclo di vita, e di strumenti "*domain-specific*", quali ad esempio simulatori per uno specifico dominio applicativo.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Al termine del corso, lo studente deve dimostrare di aver acquisito familiarità con tutte le fasi del ciclo di vita di un sistema basato su AI, dall'analisi dei requisiti, alla progettazione, al test, alla valutazione e assicurazione della qualità. Per ciascuna fase, lo studente acquisirà conoscenze e competenze relative ai processi, alle tecniche, ai linguaggi e agli strumenti necessari all'ingegnerizzazione.

Capacità di applicare conoscenza e comprensione

Al termine del corso, lo studente deve dimostrare di saper integrare competenze e soluzioni di intelligenza artificiale (ed, in particolare, *machine learning -ML*) nei processi di ingegnerizzazione dei sistemi basati su AI. Il processo formativo è orientato a fornire le competenze per utilizzare gli strumenti necessari a progettare, testare, valutare e migliorare la qualità dei sistemi basati su AI.

PROGRAMMA-SYLLABUS

Part I. Introduzione.

Concetti introduttivi. *Learning-Enabled Autonomous Systems, Cyber-Physical Systems, Self-Adaptive Systems*. Esempi di applicazione. Ciclo di vita dei sistemi intelligenti. Analisi dei requisiti, design, sviluppo, V&V, *quality assessment,* evoluzione/manutenzione. Monitoraggio a *runtime*, tolleranza ai guasti.

Part II. Analisi e Progettazione.

System-level design. Modellazione dei requisiti di sistema. *Model-based System and Software Engineering.* Simulazione, Co-simulazione. Linguaggi di modellazione (SysML, UML). Linguaggi di modellazione domain-specific (DSML) per *self-adaptiveness*. Approcci agili, DevOps, DevSecOps, ModDevOps.
Model-level (AI/ML component) design. Richiami di ML. Il workflow ML. MLOps. Progettazione, sviluppo del modello e fase operazionale. *Data Engineering pipeline; Model Engineering pipeline, deployment, monitoring, logging; Code Engineering pipeline*.

Part III. Test.

Test di modelli di ML, verifica e validazione. Test di Deep Neural Networks **(**DNN). Test black-box, test combinatoriale. Test white-box, criteri di adeguatezza. Il problema dell'oracolo. *Multiple-implementation testing*. Pseudo-oracoli. *Metamorphic testing* e *mutation testing*. *Operational testing* per stima dell'accuratezza di DNN. *Generative Adversarial Networks: Adversarial testing*. Test di Bot (*social testing*). Cenni a formal verification di DNN. Test di sistemi AI, test di integrazione, test di accettazione. Test *model-based*. Test di conformità.

Part IV. Quality assurance.
Attributi di qualità di sistemi AI. Dependability, security, privacy. Interpretability, explainability. Analisi delle prestazioni. Attributi self-*. Tecniche di *Fault avoidance, Fault tolerance, Fault removal* nel ML. Stima della qualità *model-based*, *measurement-based*. Certificazione di *safety* dei sistemi AI. *Diverse redundancy, runtime monitoring.* Stima dell'affidabilità.

Esempi/casi di studio di progettazione e test nei domini applicativi. Sistemi a guida autonoma; velivoli senza conducente; bot usati nel *customer service*, nei social media, nella realtà virtuale/metaverso.

MATERIALE DIDATTICO

LIBRO DI TESTO

Engineering Intelligent Systems - Systems Engineering and Design with Artificial Intelligence, 2022, Barclay R. Brown, Wiley. ISBN: 978-1-119-66563-2

TRASPARENZE DALLE LEZIONI ED ESERCITAZIONI disponibili sul sito web docenti di Ateneo sulla piattaforma Microsoft Teams.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà: a) lezioni frontali per circa l'70% delle ore di lezione totali, b) esercitazioni guidate per circa il 30% delle ore di lezione totali.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

L'esame prevede lo svolgimento di un elaborato progettuale ed in una prova orale.

b) Modalità di valutazione:
L'elaborato progettuale e la prova orale sono orientati ad accertare le conoscenze e le competenze acquisite in tutte le fasi del ciclo di vita di un sistema basato su AI. La valutazione è pertanto proporzionale al grado di conoscenza e capacità di applicazione di quanto appreso in ogni fase.

SCHEDA DELL'INSEGNAMENTO (SI)

"INFORMATION RETRIEVAL SYSTEMS"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ANTONIO M. RINALDI
TELEFONO: 0817683911
EMAIL: ANTONIOMARIA.RINALDI@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO: II
SEMESTRE: II
CFU: 6

INSEGNAMENTI PROPEDEUTICI

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

L'obiettivo del corso è quello di fornire agli studenti nozioni specialistiche per affrontare e risolvere le problematiche legate al processo di Information Retrieval (IR). Il corso presenterà metodi, modelli e tecniche per la ricerca di informazioni e descriverà aspetti tecnologici ed architetturali dei sistemi per l'IR. Verranno presentate metodologie per la rappresentazione delle informazioni sia testuali che multimediali tramite appositi descrittori, utilizzate tecniche di data mining, machine learning e deep learning per la loro analisi e tecnologie di memorizzazione basate su BigData in applicazioni di IR con particolare riferimento al semantic web, al web of data e ai dispositivi intelligenti. Verranno utilizzati strumenti software per la realizzazione completa di un sistema di Information Retrieval e presentati casi studio su applicazioni emergenti.

RISULTATI DI APPRENDIMENTO ATTESI

Conoscenza e capacità di comprensione

Il percorso formativo intende fornire agli studenti le conoscenze e gli strumenti metodologici necessari per analizzare le problematiche legate all'analisi delle informazioni per permettere il loro reperimento. Tali strumenti consentiranno agli studenti di comprendere le principali relazioni che sussistono tra rappresentare le informazioni, analizzarle e gestirle e di cogliere le conseguenze in termini di efficacia ed efficienza nel contesto dell'intero processo di Retrieval.

Capacità di applicare conoscenza e comprensione

Il percorso formativo è orientato a trasmettere le capacità e gli strumenti metodologici e operativi necessari ad applicare concretamente le conoscenze legate all'analisi delle informazioni nel processo di Retrieval per lo sviluppo di tecniche e l'utilizzo di tecnologie adeguate.

PROGRAMMA-SYLLABUS

INTRODUZIONE ALL'INFORMATION RETRIEVAL (IR) - Il Problema del Retrieval delle Informazioni, Data retrieval ed Information retrieval, Utenti, Sistemi di IR, Il WEB, La Rilevanza delle Informazioni, Interfacce di Ricerca, Navigazione e Ricerca, Specifica delle Query, visualizzazione dei Risultati.
MODELLI DI RETRIEVAL - Modellazione e Ranking, Modello Booleano, Modello Vettoriale, Modello Probabilistico, Modelli Avanzati per l'Information Retrieval.
VALUTAZIONE DEL PROCESSO DI RETRIEVAL - Il Paradigma di Cranfield, Precision/Recall, F-measure, Altre Misure per la Valutazione, Collezionidi Documenti.
RELEVANCE FEEDBACK E QUERY EXPANSION - L'Utente nel Processo di Retrieval, Un Framework per i Metodi di Feedback, Relevance Feedback Esplicito, Relevance feedback Implicito, Query Expansion.
DOCUMENTI E QUERY - Formati Documentali, Proprietà del Testo, Preprocessing Documentale, Thesauri, Ontologie, Text Mining, Descrittori Multimediali, Linguaggi per le Query, Proprietà delle Query, Multimedia Query.
CLASSIFICAZIONE DEL TESTO - Caratterizzazione della Text Classification, Algoritmi non Supervisionati, Algoritmi Supervisionati, Metriche di Valutazione, Organizzazione delle Classi.
MULTIMEDIA RETRIEVAL - Definizione di Multimedia, Il Gap Semantico, Image Retrieval, Audio e Music Retrieval, Video Retrieval (cenni), Multimodal Information Retrieval ed Information Fusion, Tecniche di Intelligenza Artificiale, BigData e Deep Learning per il Retrieval e la Classificazione Multimediale.
INDEXING - Il processo di Indicizzazione, Indici Inversi, Ricerca, Ranking, Costruzione, Query Strutturali. WEB RETRIEVAL - Il Web, La struttura del Web Graph, Modellare il Web, Link Analysis, Architetture dei Motori di Ricerca, Ranking nei Motori di Ricerca, Semantic Web, Web of Data, Search EngineUser Interaction.
WEB CRAWLING - Applicazioni di un Web Crawler, Architettura di un Crawler, Algoritmi di Scheduling.

MATERIALE DIDATTICO

Libro di testo:

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval: The Concepts and Technology Behind Search 2ed, Addison Wesley, 2011.

Slides del Corso.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Lezioni frontali, esercitazioni, seminari (Intelligenza Artificiale, Tecnologie di BigData e Deep Learning per Applicazioni Avanzate di IR), utilizzo di software specialistici (Lucene, Apache Solr, LIRE, OPENCV)

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità d'esame:

L'esame si articola in prova	
discussione di elaborato progettuale	1
orale	1

La preparazione verrà valutata tramite la discussione di un elaborato pratico (descrizione delle tecniche e sviluppo codice) proposto dallo studente e concordato con il docente su un argomento specifico del corso. Dopo la discussione dell'elaborato si svolgerà un orale sulla parte teorico/metodologica dell'insegnamento. Le prove concorrono in maniera paritetica al voto finale.

b) Modalità di valutazione

SCHEDA DELL'INSEGNAMENTO (SI)

"BIG DATA ENGINEERING"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: VINCENZO MOSCATO
TELEFONO: 0817683835
EMAIL: VINCENZO.MOSCATO@UNINA.OT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): II
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Scopo dell'insegnamento è quello di fornire in forma specialistica le caratteristiche peculiari delle Architetture per il Big Data Processing e la Big Data Analytics, con riferimento alla progettazione di sistemi per la gestione dati di grandi dimensioni e complessi, e all'implementazione dei processi di modellizzazione, acquisizione, condivisione, analisi e visualizzazione delle informazioni presenti nei Big Data. In particolare, verranno approfonditi le tecnologie, i framework ed i tool più diffusi per la memorizzazione, elaborazione ed analisi dei Big Data, fornendo allo studente tutte le conoscenze necessarie sia allo sviluppo di applicazioni pratiche per l'applicazione dei Big Data alla cosiddetta XInformatics, sia alla comprensione e gestione delle relative infrastrutture tecnologiche. Verranno infine trattate le problematiche ingegneristiche relative alla realizzazione di analytics su massive dataset sia per applicazioni batch sia real-time, con riferimento a quelle che sono le applicazioni emergenti dei Big Data (es. Social Network Analysis, Cyber-Security, Smart Cities, etc.).

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il percorso formativo intende fornire agli studenti le conoscenze specialistiche e tutti gli strumenti metodologici e tecnologici per la gestione di Big Data, con particolare riferimento alle caratteristiche delle moderne architetture informatiche per il Big Data Processing e la Big Data Analitycs. In particolare, tali strumenti consentiranno agli studenti, a seconda del tipo di applicazione, da un lato, di sapere definire quelli che sono i processi più adeguati di modellizzazione, acquisizione, condivisione, analisi e visualizzazione delle informazioni relative ai Big Data, dall'altro di individuare i requisiti hardware e software dei sistemi informatici di supporto.

Capacità di applicare conoscenza e comprensione

Il percorso formativo è orientato a trasmettere le capacità e gli strumenti metodologici ed operativi necessari ad applicare concretamente le conoscenze relative alla gestione di Big Data in contesi applicativi reali. In particolare, lo studente deve dimostrare di essere in grado di progettare e implementare in forma prototipale, sfruttando i framework tecnologici ed i tool esistenti, sistemi per il Big Data Processing e la Big Data Analitycs in grado di supportare differenti applicazioni sia batch sia real-time.

PROGRAMMA-SYLLABUS

1. **Introduzione ai Big Data**: Definizione e caratteristiche di un Sistema Big Data.
2. **L'ecosistema Hadoop**: HDFS, Hadoop, Yarn.
3. **Analisi di Big Data**: Pig, Hive, Giraph, Spark, GraphX. MLib.
4. **Basi di Dati NoSQL**: Key-value - Column-family, Graph, Document Database systems. Proprietà BASE vs transazioni. Teorema CAP. Cenni alle Basi di Dati NewSQL.
5. **Big Data Analytics (BDA)**: BDA Lifecycle: knowledge discovery in database, data preparation, model planning, model building, data visualization.
6. **Analisi real-time di Big Data**: Architetture Lambda e Kappa, Apache Flume, Apache Kafka, Spark Streaming.
7. **Applicazioni emergenti dei Big Data**: Social Network Analysis, Cyber-Security, Smart-Cities.
8. **Servizi Cloud per Big Data**: AWS, Microsoft Azure.

MATERIALE DIDATTICO

- Appunti del docente, Lucidi, Articoli scientifici.
- Manuali dei vari tools presentati.
- Libri di testo consigliati:
    - "Big Data: Architettura, tecnologie e metodi per l'utilizzo di grandi basi di dati",

- A. Rezzani, APOGEO, 2013. “Business intelligence. Modelli matematici e sistemi per le decisioni”,
- C. Vercelli, MacGraw-Hill Companies,2006 “Mining of Massive Datasets”, J. Leskovec, A. Rajarman, J.D.Ullman, 2014 (on line book)

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà lezioni frontali per circa il 60% delle ore totali, ed in aggiunta esercitazioni al calcolatore, sia assistite sia personali, per approfondire praticamente gli aspetti teorici ed i tool (software specialistico) introdotti, e seminari di approfondimento per le rimanenti ore. Il tutto sarà supportato da materiale didattico multimediale disponibile on-line.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	
discussione di elaborato progettuale	X
Altro	

In caso di prova scritta i quesiti sono (*)	A risposta multipla	
	A risposta libera	
	Esercizi numerici	

(*) È possibile rispondere a più opzioni

Ogni studente dovrà sviluppare durante il corso 2 Homeworks, in aggiunta all’elaborato progettuale finale, comprendenti l’uso delle tecniche e de tools presentati.

b) Modalità di valutazione:

Durante la discussione dell’elaborato progettuale saranno richiesti approfondimenti teorici relativi a parti dell’elaborato stesso, ed il tutto concorrerà alla valutazione finale.

SCHEDA DELL'INSEGNAMENTO (SI)

"DISTRIBUTED SYSTEMS AND IOT"

SSD ING-INF/05 *

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: STEFANO RUSSO
TELEFONO: 0817683832
EMAIL: STEFANO.RUSSO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE): NO
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): I
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Obiettivo dell'insegnamento è di fornire la conoscenza degli algoritmi avanzati che risolvono problemi classici nell'ingegnerizzazione dei sistemi software e dei sistemi *embedded* distribuiti (sincronizzazione dei clock, consenso; stato globale; comunicazioni di gruppo; mutua esclusione; elezioni; transazioni; tolleranza ai guasti; consistenza dei dati), nonché quelli adoperati nei sistemi *peer-to-peer*, nei sistemi IoT e nei sistemi basati sulla tecnologia blockchain.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare: di comprendere i problemi teorici (non tecnologici) alla base dei sistemi distribuiti in applicazioni affidabili, critiche, tolleranti ai guasti e/o scalabili; di conoscere i limiti teorici in funzione delle assunzioni temporali e sui fallimenti; di conoscere gli algoritmi per la loro risoluzione.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di saper astrarre le problematiche non tecnologiche della ingegnerizzazione di un sistema distribuito - sia esso di tipo software per applicazioni in rete, o di tipo *embedded* in applicazioni industriali - di saperle ricondurre a problemi teorici noti, e di saper individuare un adeguato approccio ingegneristico alla pratica risoluzione.

PROGRAMMA-SYLLABUS

Caratterizzazione dei sistemi distribuiti. Definizioni. Modelli architetturali. Modello dei guasti. Requisiti di progetto: prestazioni, qualità del servizio, scalabilità, correttezza, sicurezza, tolleranza ai guasti. Classificazione: sistemi distribuiti sincroni, asincroni, parzialmente sincroni. (0,10 CFU)

Tempo e sincronizzazione nei sistemi distribuiti. *Clock*, *skew*, *drift*, tempo astronomico, Coordinated Universal Time (UTC). Sincronizzazione dei *clock*; algoritmo di Berkeley; algoritmo di Cristian. Network Time Protocol (NTP). Relazione *happened-before*. Orologi logici di Lamport. Orologi vettoriali. (0,5 CFU)

Stato globale di un sistema distribuito. Tagli consistenti e non consistenti. Algoritmo di *snapshot* di Chandy e Lamport. Raggiungibilità dello *snapshot*. Esempi di applicazione. (0,4 CFU)

Consenso nei sistemi distribuiti. Il problema del consenso, dei generali bizantini, della consistenza interattiva. Algoritmo di Dolev. Algoritmo Paxos. *Failure detectors;* classificazione. algoritmo *Lazy FD*. Consenso con *failure detectors*. (1,25 CFU)

Comunicazioni di gruppo. Multicast affidabili e/o ordinati. Algoritmi per la loro implementazione. (0,5 CFU)

Coordinazione distribuita. Mutua esclusione distribuita; algoritmi: del server centrale, ad anello, di Ricart e Agrawala, di Maekawa. Elezioni distribuite. Algoritmo: ad anello, del prepotente. Valutazione degli algoritmi. (0,5 CFU)

Azioni atomiche distribuite. Transazioni. Proprietà ACID. Problemi di consistenza dei dati, serializzabilità, effetto domino. Controllo di concorrenza: *Two-phase lock. Two-version lock*. Transazioni composte. Transazioni distribuite, *two-phase commit*, controllo di concorrenza nelle transazioni distribuite. *Deadlock* distribuito. La memoria stabile. (0,75 CFU)

Replicazione e tolleranza dei guasti nei sistemi distribuiti. *Dependability*. La catena *fault-error-failure*. Replicazione attiva e passiva. Strategie di aggiudicazione. L'approccio *state machine replication*. (0,25 CFU)

Consistenza dei dati nei sistemi distribuiti. Replicazione e consistenza. Consistenza stretta, linearizzabile, sequenziale, causale, eventuale, *processor consistency*, *slow memory*, modelli client centrici. Teorema CAP. (0,5 CFU)

Blockchain e IoT. Blockchain: algoritmi base. *Smart contracts. Bitcoin, Ethereum, Hyperledger Fabric.* Internet delle Cose *(IoT).* Aspetti di sicurezza in IoT. Applicazioni di blockchain all'IoT. (1 CFU)

Sistemi *peer-to-peer*. Motivazioni, requisiti, applicazioni, classificazione. Overlay networks, Overlay routing, Distributed Hash Table (DHT). Esempi: Napster, Gnutella, BitTorrent, Chord, Pastry. (0,25 CFU)

MATERIALE DIDATTICO

Libri di testo:

- Coulouris, Dollimore, Kindberg, "Distributed Systems: Concepts and Design – 5th ed.", Addison-Wesley, 2011;
- Kshemkalyani, Singhal, "Distributed Computing: Principles Algorithms and Systems", Cambridge Univ. Press, 2008.

Trasparenze delle lezioni (disponibili sul sito web docente).

Pubblicazioni scientifiche:

- L. Lamport, "Time, Clocks and the Ordering of Events in a Distributed System". Communications of the ACM, Vol. 21, N. 7, Luglio 1978.
- L. Lamport, "Paxos Made Simple", 2001.
- F.B. Schneider, "Implementing Fault-Tolerant Services Using the State Machine Approach: A tutorial". ACM Computing Survey, Vol. 22, N. 4, Dicembre 1990.
- A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems". Proc. 18th IFIP/ACM Int. Conf. on Distributed Systems Platforms (Middleware 2001), Nov. 2001.
- M. Crosby, Nachiappan, P. Pattanayak, S. Verma, V. Kalyanaraman, "BlockChain Technology", 2015, https://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

La didattica è erogata:
a) per l'80% con lezioni frontali;
b) per il 10% con seminari sulle tematiche emergenti di blockchain e delle sue applicazioni all'Internet of Things;
c) per il 10% con esercitazioni guidate in aula per approfondire praticamente aspetti relativi agli orologi logici e vettoriali e alla realizzazione degli algoritmi trattati.

Gli argomenti delle lezioni frontali e dei seminari sono esposti con l'ausilio di trasparenze dettagliate, messe a disposizione dello studente nel materiale didattico tramite il sito web ufficiale del docente.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità d'esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	
altro	

L'esame si articola in una prova solo orale, consistente in tre domande su altrettanti problemi/algoritmi esposti al corso. A scelta dello studente, una delle tre domande può essere sostituita dall'esposizione del contenuto di una delle pubblicazioni scientifiche elencate nel materiale didattico.

b) Modalità di valutazione:

SCHEDA DELL'INSEGNAMENTO (SI)

"HIGH-PERFORMANCE AND QUANTUM COMPUTING"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ALESSANDRO CILARDO
TELEFONO: +39 081 7683852
EMAIL: ALESSANDRO.CILARDO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): I
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

L'insegnamento si propone di fornire agli studenti le nozioni specialistiche relative alle architetture degli odierni calcolatori impiegati per il calcolo ad alte prestazioni, approfondendo concetti legati alla struttura interna dei processori superscalari, per poi estendere la trattazione ai calcolatori multi- e many-core. A riguardo, l'insegnamento copre anche le architetture di calcolo eterogenee, in particolare basate su GPU, ed i relativi modelli di programmazione. La trattazione delle architetture di calcolo avanzate è poi estesa alle emergenti tecnologie di Quantum Computing, introducendo gli studenti a tematiche di progetto e gestione di sistema ed aspetti di programmazione.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il percorso formativo, basato su lezioni teoriche ed esercitative, punta a dotare lo studente delle conoscenze relative alle architetture degli odierni calcolatori impiegati per il calcolo ad alte prestazioni (high performance computing, HPC), con riferimento alle diverse forme di parallelismo offerte alle applicazioni. L'insegnamento pertanto approfondisce la comprensione della struttura interna dei processori superscalari, per poi estendere la trattazione ai calcolatori multi- e many-core. La prima parte del programma è anche rivolta alla presentazione delle architetture di calcolo eterogenee, in particolare basate su GPU, un paradigma assestato per lo sviluppo di applicazioni parallele ad alte prestazioni. La seconda parte del corso è invece volta ad approfondire le opportunità offerte dalle emergenti tecnologie di Quantum Computing (QC) nella prospettiva del calcolo ad alte prestazioni, affrontando tematiche di progetto e gestione di sistema ed aspetti di programmazione.

Capacità di applicare conoscenza e comprensione

Il percorso formativo è orientato a trasmettere le capacità e gli strumenti metodologici/operativi necessari ad applicare le conoscenze in termini di sviluppo di applicazioni parallele, programmazione per architetture di calcolo avanzate ed emergenti, analisi di aspetti prestazionali. Nello specifico, la parte esercitativa del corso è dedicata alla presentazione di modelli di programmazione per GPU, in particolare CUDA ed OpenCL. Inoltre, essa fornisce spunti relativi alle emergenti tecnologie di Quantum Computing, focalizzandosi su piattaforme sperimentali di QC attualmente rese disponibili da realtà industriali internazionali. In tal senso, il corso prevede la presentazione di casi di studio reali, in parte sviluppati in forma interattiva con gli studenti, con riferimento sia alle architetture parallele convenzionali sia agli emergenti scenari del QC.

PROGRAMMA-SYLLABUS

Il corso è organizzato in due tipologie di contenuti, una teorica ed una applicativa.

Parte teorica

- Architetture superscalari, esecuzione out-of-order, hardware multi-threading, etc.
- Organizzazione della memoria in sistemi paralleli: problematiche di coerenza e consistenza
- Interconnessioni a livello sistema e network on-chip
- Estensioni vettoriali: concetti generali e casi di studio relativi a processori reali
- Graphics Processing Unit: aspetti architetturali e casi di studio reali
- Modelli di programmazione per GPU: approfondimenti su concetti ed aspetti avanzati di programmazione
- Quantum Computing: aspetti tecnologici e modelli computazionali

Parte applicativa
Per il laboratorio, gli studenti svilupperanno individualmente un elaborato che prevedrà l'approfondimento verticale di aspetti tecnologici su architetture di calcolo avanzate o emergenti (QC), o alternativamente lo sviluppo ottimizzato e la presentazione di un programma su un'architettura GPU o un'estensione vettoriale.

MATERIALE DIDATTICO

- J. Hennessy, D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th Edition, Morgan Kaufmann, 2019
- R.S. Sutor, *Dancing with Qubits: How quantum computing works and how it can change the world*, Packt Publishing, 2019
- NVIDIA, *CUDA C Programming Guide*, v. 11.1.0, online, NVIDIA 2020
- V. Silva, *Practical Quantum Computing for Developers*, Apress, 2018
- D. Kaeli, P. Mistry, *Heterogeneous Computing with OpenCL 2.0*, Morgan Kaufmann, 3rd edition, 2015
- D. J. Sorin, M. D. Hill, and D. A. Wood, *A Primer on Memory Consistency and Cache Coherence*, Morgan Claypool 2011
- N. E. Jerger, T. Krishna, and L.-S. Peh, *On-Chip Networks*, Morgan Claypool, 2nd edition, 2017
- Manuali e dispense fornite durante il corso.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà:
a) lezioni frontali per circa il 65% delle ore totali,
b) esercitazioni per approfondire praticamente aspetti teorici per 8 ore,
c) laboratorio per approfondire le conoscenze applicate per 6 ore,
d) seminari per approfondire tematiche specifiche per 2 ore.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	
discussione di elaborato progettuale	X
altro	

In caso di prova scritta i quesiti sono (*)	A risposta multipla	
	A risposta libera	
	Esercizi numerici	

b) Modalità di valutazione:

COURSE DETAILS

"REAL-TIME SYSTEMS AND INDUSTRIAL APPLICATIONS"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

GENERAL INFORMATION – TEACHER REFERENCES

DOCENTE: MARCELLO CINQUE
TELEFONO: 0817683874
EMAIL: MARCELLO.CINQUE@UNINA.IT

GENERAL INFORMATION ABOUT THE COURSE

CORSO INTEGRATO (SE APPLICABILE):
MODULO (SE APPLICABILE):
CANALE (SE APPLICABILE):
ANNO DI CORSO (I, II, III): II
SEMESTER (I, II): II
CFU: 6

REQUIRED PRELIMINARY COURSES (IF MENTIONED IN THE COURSE STRUCTURE "ORDINAMENTO")

No

PREREQUISITES (IF APPLICABLE)

Basic knowledge on operating systems and programming, acquired during the bachelor's degree.

LEARNING GOALS

The course provides students with advanced notions related to real-time systems and their application in different industrial areas, with particular reference to mission and safety-critical systems. It provides the necessary skills for designing and developing real-time mixed-criticality systems, using operating systems and virtualization platforms for real-time embedded systems, including hybrid high-performance hardware architectures. The course addresses both the recommendations imposed by certification standards in different industrial contexts, such as automotive, railways and avionics, and research initiatives on related themes, such as the Industrial Internet of Things and Industry 4.0.

EXPECTED LEARNING OUTCOMES (DUBLIN DESCRIPTORS)

Knowledge and understanding

The student needs to show to understand the theoretical problems at the base of real-time computing, to know the scheduling and resource management algorithms to address such problems. He/she has to be able to illustrate the virtualization principles and techniques that can be used to realize mixed-criticality systems while respecting isolation properties, to recognize the main industrial areas in which such systems are used, along with the constraints imposed by the normative.

Applying knowledge and understanding

The student needs to show to be able to apply the methodological instruments for the feasibility analysis and the dimensioning of real-time systems on single-core, multi-core and/or asymmetric/hybrid platforms; to be able to analyze the theoretical and practical problems concerning the use of operating systems and virtualization platforms in embedded and industrial environments, to be capable to design and implement complex real-time mixed-criticality systems, with reference to processes and technologies conformant to the main industrial standards.

COURSE CONTENT/SYLLABUS

Introduction to real-time systems. Fundamental properties. Hard and soft real-time. The safety concept, safety- critical systems. Scheduling of periodic tasks: cyclic executive, rate monotonic (RM), earliest deadline first (EDF). Aperiodic servers: polling server, deferrable server, sporadic server, total bandwidth server, hard constant bandwidth server. Resources management: non-preemptive protocol, highest locker priority, priority inheritance and priority ceiling. Feasibility tests based on upper bounds and on response time analysis.

Real-time kernel architectures. Real-time operating systems. Latency sources in computing platforms. Real-time executive. Monolithic kernels. Microkernels and L4 family. Dual kernels: RTAI and Xenomai examples. Preemptable kernels. The PREEMPT_RT patch for the Linux kernel.

Programming real-time tasks in Linux. The RT-POSIX standard. Programming cyclic periodic tasks. Fixed priority scheduling in Linux (SCHED_FIFO). Programming real-time applications on Xenomai. Real-time performance evaluation. Practical applications.

Real-time systems monitoring. Monitoring methodologies: embedded constraints and monitored constraints. Fault tolerance and timing failures management. Rule-based logging. Error propagation analysis.

Real-time multi-processing. Use of multiprocessing systems for real-time computing. Tasks models and theoretical limits. Symmetric and Asymmetric Multi-Processing. Partitioned scheduling of sporadic tasks with EDF and RM. Bin packing allocation. Global scheduling: the pfair algorithm, global EDF, Dhall effect, pfEDF, RM-light and RM-US. Feasibility bounds. Interference and isolation in multiprocessors due to memory hierarchies.

Mixed-criticality systems. The Vestal's model. Multi-criticality response time analysis. Audsley method. Hierarchical scheduling: open system architecture, fixed-priority hierarchical systems, hierarchical constant bandwidth server. The use

of hybrid asymmetric platforms and multi-processor systems on chip for the realization of mixed-criticality systems. OpenAMP: RemoteProc, RPMsg and examples of use on the Zync Ultrascale+ platform.

Real-time virtualization. Introduction to virtualization: CPU, memory and I/O virtualization. Virtualization issues in real-time systems. Types of hypervisors. Examples of real-time hypervisors: XEN with null and RTDS schedulers, Jailhouse and practical examples of use. Real-time cloud and real-time containers: architectural alternatives and examples of implementation/use in dual-kernel environments.

Industrial applications. The automotive context: the ISO 26262 standard, ASIL and development process, coding rules and MISRA-C, the concept of SEooC, the OSEK standard, the AUTOSAR standard, the CAN protocol, operating systems and hypervisors used in automotive. The avionics context: the DO 178B standard, the ARINC 653 standard, IMA architecture and APEX interface, real-time networking in avionics, operating systems and hypervisors used in avionics. The railways context: the CENELEC EN 50128 standard. The Industrial Internet of Things: terminology, architecture, notes on standardization initiatives and communication protocols.

READINGS/BIBLIOGRAPHY

- Textbooks:
  - Giorgio Buttazzo: "Hard real-time computing systems: Predictable Scheduling Algorithms and Applications", Third Edition, Springer, 2011.
  - S. Baruah, M. Bertogna, G. Buttazzo. "Multiprocessor Scheduling for Real- Time Systems", Springer, 2015
- Lessons' slides, reports and scientific articles available on the official course website.

TEACHING METHODS

Teacher will use a) lectures for approx. 75% of total hours, b) practical guided exercises in classroom to deepen the understanding and skills on real-time tasks programming and the use of hybrid virtualization architectures for about 20% of total hours, and c) seminars on specific themes, such as industrial standards, for about 5% of total hours.
Lectures and seminars are explained with the help of detailed slides, which will be provided to students through the official course website.

EXAMINATION/EVALUATION CRITERIA

b) Exam type:

Exam type	
written and oral	
only written	
only oral	X
project discussion	X
other	

The exam is articulated in a single oral session, which comprehend the presentation of a project assigned during the course and three questions on problems, algorithms and theoretical/technological solutions among the ones explained at the course. The project can be developed in teams of maximum three students.

SCHEDA DELL'INSEGNAMENTO (SI)

"EMBEDDED SYSTEMS"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ALESSANDRO CILARDO
TELEFONO: +39 081 7683852
EMAIL: ALESSANDRO.CILARDO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): II
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

L'insegnamento si propone di fornire agli studenti le nozioni specialistiche relative ai moderni *Sistemi Embedded* (sistemi informatici dedicati espressamente progettati ed integrati in forma "embedded" in specifici prodotti, spesso vincolati a soddisfare taluni requisiti di tipo real-time e prestazionali, oltre che requisiti su consumi, ingombro, affidabilità e sicurezza) quali quelli ampiamente impiegati nei sistemi industriali (trasporti: automotive e ferroviario; energia; sistemi di controllo meccanici) o di largo consumo (telefonia, intrattenimento, elaborazione multimediale) o, in generale, sistemi dell'internet delle cose (Internet of Things, IoT), di e-health, robotica ed Intelligenza Artificiale.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il percorso formativo, basato su lezioni teoriche ed esercitative, è volto a dotare lo studente delle conoscenze relative alle architetture di tipo embedded, con enfasi sugli aspetti tecnologici e metodologici, così come della comprensione dei flussi di progettazione per sistemi di tipo System-on-Chip (SoC) e Multi-Processor System-on-Chip (MPSoC). Il corso presenta, tra gli altri, sistemi SoC con processore ARM su schede STM32F4 della STMicroelectronics ed altri processori per applicazioni embedded sintetizzati su FPGA. Nelle lezioni di laboratorio, sono anche sviluppate conoscenze di alcuni ambienti IDE per la progettazione, composti da compilatori di linguaggi HDL, quali VHDL e Verilog, debugger, simulatori e strumenti per il mapping tecnologico.

Capacità di applicare conoscenza e comprensione

Il percorso formativo è orientato a trasmettere le capacità e gli strumenti metodologici/operativi necessari ad applicare le conoscenze in termini di progettazione ingegneristica di sistemi embedded, basati su differenti architetture di tipo System-on-Chip (SoC), Multi-Processor System-on-Chip (MPSoC) e speciali (DSP, hardware dedicato), sintetizzate anche con componenti FPGA già studiati in insegnamenti precedenti del percorso di laurea magistrale. La progettazione farà ricorso a metodologie di sviluppo e ad ambienti IDE professionali ampiamente impiegati nel mondo industriale. La parte applicativa, in particolare, prevede lo sviluppo di un progetto d'aula.

PROGRAMMA-SYLLABUS

Il corso è organizzato in tre parti principali: una teorica, una tecnologica ed una applicativa.

Parte teorica

- Architetture speciali e generali dei Sistemi Embedded e loro strutturazione a livelli.
- Architetture hardware basate sull'integrazione di hardware commerciale (processori, periferiche, multiprocessori, multicomputer, DSP, system on chip, etc.), hardware specifico (progettato mediante HDL e tecnologie di sintesi automatica) e dispostivi per il collegamento con attuatori e sensori di campo.
- Caratteristiche architetturali dei processori e dei sistemi per applicazioni di tipo System-on-Chip (SoC) e Multi-Processor System-on-Chip (MPSoC).
- Software di base, sistemi operativi e software applicativo per sistemi dedicati con o senza vincoli di real-time.
- Generalità su sensori e attuatori: principi di funzionamento, tipologia, interfacce ed esempi d'uso.
- Reti di sensori: architettura e software di servizio.
- Logiche di partizionamento di un sistema dedicato tra unità funzionali realizzate in hardware e/o in software dei sistemi embedded SoC e MPSoC.
- Sistemi di elaborazione per applicazioni industriali critiche per costo, affidabilità, tempo, sicurezza, consumi e ingombro.

- Ciclo di sviluppo: metodologie e strumenti per la sintesi e il codesign.
- Sviluppo di un sistema embedded su board STMicroelectronics della serie STM32xxx con processori ARM.

Parte tecnologica

- Livelli hardware/software in cui è strutturato un sistema embedded: livello HAL, middleware, di sistema operativo RTOS e applicativo, anche con requisiti real-time, sottosistema di I/O caratterizzato dai più diffusi dispositivi di I/O, da sensori e attuatori e loro reti.
- Dispositivi di I/O sintetizzabili come IP *core* (UART/USART, PIO, I2C, SPI, USB, etc.) su FPGA.
- Componenti di base: sottosistemi di memoria, architettura di interconnessione. Bus e sistemi di interconnessione ad elevate prestazioni: i bus AMBA e AXI.
- Architetture dei più diffusi processori per sistemi embedded: la famiglia ARM; processori nativi della famiglia di FPGA (MicroBlaze di Xilinx); altri processori sintetizzabili come IP core su FPGA (processori di fascia bassa a 8, 16, 32 bit di tipo CISC e RISC, Java machine); architettura aperta RISC-V.
- Architetture System-on-Chip, famiglia di sistemi STM32Fxx.
- Architettura dei sistemi paralleli di tipo multiprocessore e reti di interconnessione.
- Architettura dei sistemi multicore e Multi-Processor System-on-Chip.
- Componenti programmabili ibridi composti da SoC o MPSoC (con processori di tipo general-purpose RISC/CISC e/o speciali quali ad esempio i Digital Signal Processor, DSP) e FPGA (per sintetizzare macchine speciali).
- Dispositivi ibridi FPGA-multicore: la famiglia Zynq.
- Architettura delle schede Digilent Zybo Zynq e ZedBoard Zynq
- Librerie di IP-Core, "intellectual Property Core", appositamente sviluppate o presenti in librerie pubbliche e commerciali.

Parte applicativa

Per il laboratorio, gli studenti, organizzati in gruppi di lavoro, svilupperanno un progetto concreto di sistema embedded, suddiviso in sottosistemi, a partire dalle specifiche funzionali loro assegnate. La progettazione farà ricorso a metodologie di sviluppo e ad ambienti IDE a supporto, concretamente impiegati in ambito industriale. In particolare, si farà riferimento ad esempi di sistemi "on board" commerciali messi a disposizione degli studenti, con processori della famiglia ARM ed FPGA delle famiglie Xilinx.

MATERIALE DIDATTICO

- G. Conte, A. Mazzeo, N. Mazzocca, P. Prinetto, "Architettura dei calcolatori", Cittastudi edizioni, 2015.
- F.Vahid, T.Givaris, "Embedded System Design - A unified Hardware/Software Introduction", Wiley, 2002.
- K. Papovic, F. Rousseau, A. A. Jerraya, M. Wolf, "Embedded Software Design and Programming of Multiprocessor System-on-Chip", Springer, 2010.
- M. Domeika, Software Development for Embedded Multi-core Systems a Practical Guide Using Embedded Intel ® Architecture
- "Automotive Embedded Systems Handbook, Industrial Information Technology Series CRC Press, 2009, N. Navet, F. Simonot Lion eds.
- K. Yaghmour, J. Masters, G. Ben-Yossef, and P. Gerum, "Building Embedded Linux Systems", O'Reilly.
- Wayne Wolf, "Computers as components", Morgan Kaufman, 2000. Planned 2nd ed
- P. Marwedel,"Embedded System Design", Kluwer academic publishers, ISBN: 1-4020-7690-8, November 2003
- J. Sauermann, M. Thelen, "Realtime Operating Systems Concepts and Implementation of Microkernels for Embedded Systems",
- Manuali e datasheet dei vari dispositivi utilizzati e forniti durante il corso.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà:
a) lezioni frontali per circa il 55% delle ore totali,
b) esercitazioni per approfondire praticamente aspetti teorici per 8 ore,
c) laboratorio per approfondire le conoscenze applicate per 10 ore,
d) seminari per approfondire tematiche specifiche per 4 ore.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	
discussione di elaborato progettuale	**X**
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

b) Modalità di valutazione:

SCHEDA DELL'INSEGNAMENTO (SI)

"WEB AND REAL TIME COMMUNICATION SYSTEMS"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALE - DOCENTE

DOCENTE: SIMON PIETRO ROMANO
TELEFONO: +39 081 7683823
EMAIL: SIMONPIETRO.ROMANO@UNINA.IT

INFORMAZIONI GENERALI – ATTIVITA'

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): II
CFU: 6

REQUIRED PRELIMINARY COURSES (IF MENTIONED IN THE COURSE STRUCTURE "ORDINAMENTO")

No

PREREQUISITES (IF APPLICABLE)

No

LEARNING GOALS

The aim of the class is to provide students with advanced notions in the field of both web-based and real-time communication. Students will become familiar with the design and development of complex communication systems, by focusing both on the client-side and on the server-side. They will become acquainted with the networking protocols that form the basis of web-based and real-time communication, as well as learn how to use the standard Application Programming Interfaces (APIs) laying on top of them. This is both a programming class and a networking class. All of the networking protocols and solutions will be first studied in detail and then put into practice through the design and implementation of proof-of-concept prototypes representing real-world application scenarios.

EXPECTED LEARNING OUTCOMES (DUBLIN DESCRIPTORS)

At the end of the class, students will be able to demonstrate advanced knowledge in the field of web-based and real-time communication. They will have a clear understanding of the protocols and APIs representing the state of the art in both fields. Step by step, the class will bring them towards an integrated approach, whereby web-based and real-time communication eventually co-exist in a single, unified scenario adhering to the latest standards issued by both the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C).

They will be ready to enter the professional arena and become part of cutting-edge development teams, by actively contributing to the design and the implementation of advanced communication systems. Successful completion of the class will allow them to effectively apply the acquired knowledge to a number of real-world scenarios requiring advanced engineering capabilities, by putting together their networking competences and their advanced programming skills.

Knowledge and understanding

Students need to show ability to know and understand problems related to both the communication and the programming issues associated with the design and development of complex interoperating distributed systems.

They need to elaborate arguments related to the relationship between communication protocols, synchronous and asynchronous interaction, client-side, server-side and peer-to-peer programming, by embracing an engineering approach and looking at them in an integrated fashion. The class provides students with advanced knowledge in the field of web-based real-time communication, by also illustrating how to leverage both methodological and practical tools in order to design and implement effective, interoperable, scalable and secure real-time multimedia communication systems that are compliant with state-of-the-art standard protocols and APIs. Such tools will allow students to grasp the causal connections among network-based communication and event-based programming, as well as understand the implications of the adoption of an agile design and development paradigm for the realization of advanced communication systems.

Applying knowledge and understanding

Students need to show ability to design complex systems involving distributed components that exchange multiple media in a real-time fashion. They will have to demonstrate a clear understanding of the main networking protocols offering support to such systems for all what concerns their communication requirements. They will have to demonstrate advanced programming skills, with special reference to the use of the standard APIs that are offered to programmers both on the client-side and on the server-side. The course delivers skills and tools needed to apply knowledge in practice, favoring the ability to use a methodological approach that properly integrates different technologies (as well as different programming languages) in a unified framework allowing to effectively look after the many facets of a complex communication system.

COURSE CONTENT/SYLLABUS

The class will proceed step-wise towards the final goal of designing and implementing integrated systems putting together the most up-to-date solutions in the fields of web-based and real-time communication. Web-based communication protocols and technologies will be first introduced, by focusing both on the communication and on the programming aspects. Server-side web application development will be analyzed in detail, with the help of real-world examples associated with state-of-the-art frameworks and programming languages. Client-side web programming through the JavaScript language will be discussed, with a focus on both the synchronous and the asynchronous paradigm. The XML (eXtensible Markup Language) language and APIs will be introduced and analyzed in depth. The focus will then move to real-time communication, by looking at both the data layer and the signaling layer. State-of-the-art streaming technologies will be discussed in detail and comparatively analyzed. Network reachability and NAT (Network Address Translation) traversal issues will be presented, together with the standard protocols that have been designed in order to effectively deal with them. A detailed overview of the most up-to-date Instant Messaging applications and related standard protocols will be presented. Finally, the class will show how to put things all together in an integrated framework that leverages both the IETF RtcWeb and the W3C WebRTC standard protocols and APIs in order to realize next-generation web-enabled real-time communication scenarios.

Syllabus:

- Web-based communication basics: HTTP protocol deep-dive
- Server-side programming basics: Common Gateway Interface (CGI)
- Server-side programming in Java through Java servlets
- Server-side programming in Python through the Flask framework
- Server-side programming in Java through the Spring Boot framework
- Client-side programming with JavaScript
- Asynchronous JavaScript and XML (AJAX) and the Fetch API
- The XML language
- XML programming
- Server-side programming in JavaScript with Node.JS
- Web-based interaction through WebSockets
- The Real-time Transport Protocol (RTP)
- Voice over IP (VoIP) applications and the Session Initiation Protocol (SIP)
- Streaming protocols and technologies:
    - Real Time Streaming Protocol (RTSP)
    - MPEG-DASH (Dynamic Adaptive Streaming over HTTP)
    - Real Time Messaging Protocol (RTMP)
    - Peer-to-peer approaches
- NAT-traversal protocols:
    - Session Traversal Utilities for NAT (STUN)
    - Traversal Using Relays Around NAT (TURN)
    - Interactive Connectivity Establishment (ICE)
- Instant Messaging protocols and technologies:
    - Internet Relay Chat (IRC)
    - Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIP/SIMPLE)
    - eXtensible Messaging and Presence Protocol (XMPP)
    - WebRTC-enabled IM through data channels
    - Web-socket enabled IM
    - Message Queuing Telemetry Transport (MQTT) enabled IM
    - Advanced Message Queuing Protocol (AMQP) enabled IM
    - Simple Text Orientated Messaging Protocol (STOMP) enabled IM
    - The RabbitMQ library
    - Secure, decentralized real-time communication through Matrix
- User interface design and implementation through React.js
- Using native components instead of web components with React Native

- WebRTC: the new frontier of real-time communications in the web
- The Janus WebRTC media server and gateway

READINGS/BIBLIOGRAPHY

1. *Official references like, e.g., Requests For Comments (RFC), available at: https://www.ietf.org*
2. *"Real-Time Communication with WebRTC", Salvatore Loreto, Simon Pietro Romano, Released May 2014, Publisher: O'Reilly Media, Inc. ISBN: 9781449371876*
3. *Slides and additional materials provided by the teacher and made available on the official sites associated with the class*

TEACHING METHODS

Teacher will use a hands-on approach for the entire duration of the class. All of the course topics will be both presented in theory and further analyzed through practical examples.
All of the students will have to work on a practical project, either individually or in groups, focused on an in-depth study of one or more of the topics addressed during the class.

EXAMINATION/EVALUATION CRITERIA

a) Exam type:

Exam type	
written and oral	
only written	
only oral	X
project discussion	X
other	

In case of a written exam, questions refer to: (*)	**Multiple choice answers**	
	Open answers	
	Numerical exercises	

(*) multiple options are possible

b) Evaluation pattern:

- 35% of the final mark will depend on an evaluation of the practical project developed by the student (either individually or in group). The project itself will have to be delivered (with a fully-fledged documentation, including source code, if applicable) to the teacher at least seven days in advance of the oral examination date;
- 65% of the final mark will depend on the results of the oral interview.

SCHEDA DELL'INSEGNAMENTO (SI)

"DECENTRALIZED APPLICATIONS AND BLOCKCHAIN"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: GIUSEPPE ACETO
TELEFONO: 081-7683851
EMAIL: GIUSEPPE.ACETO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): I
CFU: 6

REQUIRED PRELIMINARY COURSES (IF MENTIONED IN THE COURSE STRUCTURE "ORDINAMENTO")

No

PREREQUISITES (IF APPLICABLE)

Computer Networking, Computer Programming

LEARNING GOALS

The course aims to provide the skills for understanding the functioning of the main blockchain-based platforms (both public and permissioned) and for the development of smart contracts and decentralized applications (DApps).

The course also intends to provide the methodological tools to analyze and evaluate the set of technologies available for the implementation of DApps, to guide their design, development and adoption in different application contexts.

The course includes practical and laboratory classes functional to the development of a course project.

EXPECTED LEARNING OUTCOMES (DUBLIN DESCRIPTORS)

Knowledge and understanding

The student must demonstrate knowledge and understanding of the problems arising and solutions available relating to the design and development of decentralized applications (DApps), with particular reference to blockchain-based architectures, the problem of consensus, and tools related to blockchains.
The student must also demonstrate knowledge and understanding of the concept of smart contracts, and the main languages for the programmability of blockchain-based architectures.

Applying knowledge and understanding

The student must demonstrate being able to design DApps taking into account the application context, the available technologies and their characteristics. They must also be able to develop simple DApps by effectively using the knowledge acquired regarding the programmability of blockchain-based architectures.

COURSE CONTENT/SYLLABUS

Part I - Background

Introduction and contextualization of cryptocurrencies, blockchain, smart contracts and decentralized applications.
Introduction to distributed systems, decentralized systems and the consensus problem. Distributed ledger technologies.
Fundamentals of cryptography: public key algorithms, digital signature, hashing, validation.
Consensus protocols: Proof of Work, Proof of Stake, Proof of Authority, other protocols.
Smart Contracts and Tokens.
Types of tokens, fungible and non-fungible (NFT).
Trust and communities: hard fork and soft fork.

Part II - Technologies

Blockchain Technologies: Bitcoin, Ethereum, IOTA and other major blockchains.
Scalability and consensus protocols; Layer 2 solutions: Lighting Network, Polygon.
Tools for using blockchain (wallet, chain explorer, gateway and API), off-chain systems, IPFS.
Lab #1: Use of test chains of selected blockchains

Part III - Programmability

Decentralized Applications on Blockchain (DApp).
DApp prototype development (smart contract + interface + off-chain systems).
Test-driven development of DApp.
DApps models and emerging standards.
Lab #2: Prototype implementation of DApp

Part IV - Challenges and perspectives
Security aspects related to DApps and smart contracts.
Network Security and Blockchain: Blockchain Threats and Blockchain-Based Solutions.
Blockchain privacy and blockchain interoperability (cross-chain technologies).
Performance measurement of blockchain-based technologies.
Integration of Artificial Intelligence and blockchain.
Applications of blockchain-based technologies in SDN, IoT, 5G/6G, eHealth contexts.
Decentralized Autonomous Organization (DAO).
Challenges and opportunities for the adoption and development of Blockchain and DApps.

READINGS/BIBLIOGRAPHY

- Lecture slides and notes
- Further teaching material provided by the teacher
- *Dannen, Chris. Introducing Ethereum and solidity. Vol. 1. Berkeley: Apress, 2017.*

TEACHING METHODS

The course will consist of:
a) lectures for approx. 70% of total hours;
b) laboratories and/or practical lessons for the remaining 30% of total hours.

EXAMINATION/EVALUATION CRITERIA

a) Exam type:

Exam type	
written and oral	
only written	
only oral	X
project discussion	X
other	

In case of a written exam, questions refer to: (*)	**Multiple choice answers**	
	Open answers	
	Numerical exercises	

(*) multiple options are possible

The oral exam will include the presentation and discussion of a course project previously prepared by the student.

b) Evaluation pattern:

SCHEDA DELL'INSEGNAMENTO (SI)

"WIRELESS NETWORKS AND IOT TECHNOLOGIES"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: PROF. STEFANO AVALLONE
TELEFONO: +39 081 768 3902
EMAIL: STEFANO.AVALLONE@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): II
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Scopo del corso è impartire una conoscenza approfondita delle principali tecnologie impiegate in reti wireless, sia ad alte prestazioni (Wi-Fi) che caratterizzate da vincoli sul consumo energetico dei dispositivi (LoraWAN, ZigBee), come richiesto ad esempio dal paradigma Internet of Things (IoT). Tale scopo è perseguito attraverso l'analisi delle principali problematiche affrontate dalle tecnologie wireless considerate e la presentazione delle più recenti soluzioni proposte dagli enti internazionali di standardizzazione. Il corso è focalizzato principalmente sulle problematiche relative all'accesso al mezzo, all'instradamento e al supporto per le applicazioni nelle reti wireless. Gli obiettivi formativi principali sono: la conoscenza dei principali algoritmi distribuiti per l'accesso al mezzo wireless; l'acquisizione delle principali metodologie per l'analisi delle prestazioni delle tecniche di accesso wireless; la conoscenza delle problematiche di sicurezza nelle reti wireless; la comprensione delle problematiche derivanti dalla necessità di ridurre il consumo energetico dei dispositivi; la conoscenza dei protocolli per il supporto delle applicazioni in reti IoT; la capacità di utilizzare strumenti per la simulazione di reti wireless.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Gli studenti devono dimostrare di conoscere le principali tecnologie di reti locali di tipo wireless e le principali differenze tra di esse. Gli studenti devono comprendere le problematiche relative all'accesso al mezzo wireless condiviso e capire le scelte progettuali alla base delle diverse soluzioni proposte dalle diverse tecnologie. Il percorso formativo intende fornire agli studenti le conoscenze e gli strumenti metodologici di base necessari per analizzare le prestazioni delle reti locali di tipo wireless. Gli studenti devono dimostrare di conoscere strumenti per la cattura del traffico di rete reale e strumenti per l'esecuzione di simulazioni di rete .

Capacità di applicare conoscenza e comprensione

Gli studenti devono essere in grado di identificare la tecnologia di rete locale di tipo wireless più adatta ad un determinato contesto applicativo, sulla base delle caratteristiche dei dispositivi e dell'ambiente all'interno del quale operano e sulla base dei requisiti applicativi imposti. Gli studenti devono essere in grado di selezionare le tecniche e i meccanismi messi a disposizione da una tecnologia di rete wireless al fine di raggiungere determinati obiettivi. Gli studenti devono essere in grado di analizzare le prestazioni delle diverse tecniche proposte dalle tecnologie di reti locali di tipo wireless anche mediante l'ausilio di strumenti di simulazione di rete.

PROGRAMMA-SYLLABUS

Reti WLAN: architetture, definizione dei componenti, procedure di gestione e controllo. Reti WLAN: cenni sul livello fisico (IEEE 802.11a/b/g standards). Livello MAC: DCF e PCF. La sicurezza nelle reti WLAN: meccanismi per l'autenticazione (WEP, PSK, 802.1x, SAE) e algoritmi di cifratura (WEP, TKIP, CCMP, GCMP). Estensioni per il supporto della Qualità del Servizio: lo standard IEEE 802.11e e le funzioni di accesso al mezzo EDCA e HCCA. Metodologie per la valutazione delle prestazioni delle reti WLAN basate su IEEE 802.11. Evoluzione delle reti WLAN: gli emendamenti 802.11n, 802.11ac e 802.11ax. Reti wireless ad-hoc: scenari applicativi, problematiche, protocolli di routing reattivi e proattivi. Reti wireless mesh: scenari applicativi, procedure di gestione e controllo, protocolli per la selezione dei percorsi, interoperabilità con altri segmenti di rete LAN. Tecnologie di comunicazione wireless per l'Internet of Things (IoT): LoraWAN e Zigbee (IEEE 802.15.4 standard). Protocolli applicativi per l'IoT: MQTT e CoAP. Strumenti e metodi per la simulazione di reti wireless.

MATERIALE DIDATTICO

Libro di testo consigliato:
Stefano Avallone, “Protocolli per Reti Mobili”. McGraw-Hill Italia. ISBN: 978-88-386-7414-3

Altro materiale didattico utile:
Slides delle lezioni, articoli scientifici e standard selezionati.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

La didattica verrà erogata principalmente attraverso lezioni frontali. Circa 6 ore saranno dedicate all’illustrazione del simulatore di rete ns-3, che verrà utilizzato come strumento per l’esecuzione di simulazioni di rete. La documentazione relativa al simulatore ns-3 è disponibile online sul sito web del simulatore.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

L’elaborato consiste nella realizzazione di un programma per il simulatore di rete ns-3 che abbia come obiettivo la valutazione di una o più tecniche utilizzate dalle tecnologie di rete wireless oggetto del corso. L’elaborato incide per un massimo di 3 punti sulla valutazione complessiva.

SCHEDA DELL'INSEGNAMENTO (SI)

"CLOUD PLATFORMS AND INFRASTRUCTURE-AS-CODE"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ROBERTO CANONICO
TELEFONO: +39 081 7683831
EMAIL: ROBERTO.CANONICO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): I
SEMESTRE (I, II): II
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Sistemi Operativi, Reti di Calcolatori, Programmazione, Cloud and Network Infrastructures

OBIETTIVI FORMATIVI

Scopo del corso è impartire una conoscenza approfondita delle principali metodologie e tecniche impiegate nei contesti di Cloud Computing (pubblico, privato ed ibrido) per il dimensionamento dinamico, la configurazione e la gestione di infrastrutture virtualizzate. In particolare, il corso tratta:

- gli aspetti ingegneristici di progetto e realizzazione di un moderno datacenter e le principali tecnologie IT impiegate in questo specifico contesto;
- le peculiari soluzioni architetturali e protocollari che caratterizzano il networking nel contesto dei datacenter;
- le principali tecniche di virtualizzazione attualmente impiegate per le varie componenti di un sistema IT basato sul Cloud e come queste siano impiegate per la realizzazione di sistemi scalabili, elastici, flessibili e riconfigurabili attraverso il dispiegamento e l'orchestrazione di VM, container e componenti serverless;
- le tecniche di automazione che, attraverso il paradigma DevOps, consentono di automatizzare le procedure di messa in esercizio, configurazione e gestione dei sistemi cloud e rete.

Il corso prevede anche una parte laboratoriale ed esercitativa funzionale allo sviluppo di un elaborato.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente dovrà dimostrare di aver compreso le caratteristiche fondamentali delle tecniche di virtualizzazione utilizzate nei sistemi cloud e di come esse debbano essere gestite per la realizzazione e la messa in esercizio di sistemi scalabili.

Capacità di applicare conoscenza e comprensione

Lo studente dovrà dimostrare di aver acquisito la capacità di applicare le tecniche apprese per la risoluzione di tipici problemi di progetto di applicazioni scalabili in cloud e per la configurazione delle componenti infrastrutturali di un datacenter che ospiti applicazioni cloud.

PROGRAMMA-SYLLABUS

Part I – Cloud Data Center engineering

Datacenter architecture and organization. General architecture and organization of a datacenter. Main datacenter facilities. Rack layouts. Rackable servers. Blade servers. ANSI/TIA-942 standard. Data Center Infrastructure Management (DCIM).

Data center power management. Green datacenters and green cloud computing. Cooling solutions for datacenters. Power Usage Effectiveness (PUE).

Datacenter networking. Transmission media for datacenter links. UTP cabling. Optical fibers. Optical transceivers. Access layer organization: Top-of-Rack vs. End-of-Row. Leaf-Spine datacenter networks. Multipath in datacenter networks. ECMP and TRILL. Flowlets. TCP Incast and TCP variants for datacenter networks.

Storage networking technologies. SANs. Fiber Channel. Fibre Channel over Ethernet (FcoE). iSCSI.

Datacenters for HPC applications. Infiniband.

Virtualization technologies. The virtualization concept. Different approaches to IT resource virtualization. Type-1 and Type-2 hypervisors. VM networking.

Container-based virtualization. Docker. Docker Image Layers. Container image repositories. Docker networking.

Virtual routers. Virtualized network functions. Network Function Virtualization (NFV). Virtual Network Functions chaining and orchestration.

Multi-tenancy in cloud networking. IEEE 802.1ad Q-in-Q. Overlay networking. NVGRE. VxLAN.

Lab #1: Docker. Creation of Docker containers. Multi-stage docker build.

Part II - Cloud Platforms

Modern application design in public cloud contexts.

Private Cloud platforms. OpenStack. OpenStack general architecture and fundamental services.

Resource management and configuration in private cloud infrastructures.
A comparison of IaaS and PaaS offerings by main Cloud Service Providers.
The AWS, Azure and Google public cloud offerings.
Identity management in the Cloud. Cloud APIs and interoperability.
Microservices.
Serverless computing. AWS Lambda.
Cloud edge computing.
Lab #2: OpenStack.
Labs #3, #4: A practical introduction to public cloud offerings.

Part III - Cloud and network automation
Orchestration of Docker containers. Kubernetes.
Network automation. NETCONF and RESTCONF. YANG data models.
Infrastructure as Code paradigm.
DevOps and Cloud automation in private and public cloud platforms. Basic DevOps concepts. CI/CD.
Frameworks for automatic deployment and configuration of cloud components. Ansible. YAML. Terraform.
Lab #5: Use of YAML for automatic deployment of cloud components.

MATERIALE DIDATTICO

- Dispense e lucidi delle lezioni
- Altro materiale didattico reso disponibile dal docente
- *Cloud Native DevOps with Kubernetes*, 2nd Edition, Justin Domingus, John Arundel. O'Reilly Media, 2022
- *Cloud Native Data Center Networking: Architecture, Protocols, and Tools (1st Edition).* Dinesh G. Dutt. O'Reilly, 2019
- *Network Programmability and Automation.* Jason Edelman, Scott S. Lowe, Matt Oswalt. O'Reilly, 2018

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso consisterà di: a) lezioni frontali per l'80% delle ore totali; b) esercitazioni per il restante 20% delle ore totali.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità d'esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

L'esame consisterà di una prova orale durante la quale verrà anche discusso un elaborato progettuale precedentemente redatto dallo studente.

b) Modalità di valutazione:

SCHEDA DELL'INSEGNAMENTO (SI)

"SYSTEMS SECURITY"

SSD ING-INF/05*

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: VALENTINA CASOLA
TELEFONO: 0817683907
EMAIL: VALENTINA.CASOLA@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II): II
SEMESTRE (I, II): I
CFU:6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno.

EVENTUALI PREREQUISITI

Conoscenze di programmazione; conoscenze dei principi dell'ingegneria del software.

OBIETTIVI FORMATIVI

Il corso si pone l'obiettivo di fornire un'impostazione metodologica e tecnologica per il progetto di sistemi sicuri. Il corso prevede di analizzare le tecniche di progetto standard con riferimento allo sviluppo ed uso dei principali meccanismi di sicurezza, tra cui: meccanismi di autenticazione e controllo accessi, meccanismi di sicurezza crittografici, meccanismi per la protezione delle comunicazioni e dei sistemi distribuiti. Sono inoltre presentati i principali elementi per l'analisi dei rischi e delle minacce applicabili ad un sistema per guidare le fasi di progettazione e le principali tecniche di assessment e testing della sicurezza dei sistemi.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e comprendere le problematiche relative al progetto di sistemi sicuri, con riferimento alle metodologie di analisi e progettazione, standard, presentate durante il corso, e considerando i vincoli specifici derivanti dalle tecnologie utilizzate. Deve inoltre dimostrare di comprendere le caratteristiche fondamentali di diversi meccanismi di sicurezza e di saper individuare i controlli più appropriati per soddisfare specifici requisiti di progettazione.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di eseguire l'intero ciclo di analisi, progettazione e sviluppo di un sistema sicuro, dalla fase di analisi dei rischi e delle minacce alla identificazione dei meccanismi di controllo più opportuni, alla loro implementazione e corretta configurazione, fino al testing finale della sicurezza del sistema, utilizzando strumenti e ambienti di sviluppo di ampio utilizzo.

PROGRAMMA-SYLLABUS

Course Introduction: Basic terminology, Overview of system security, Policy/mechanism separation, Security requirements.
Fundamentals of cryptography: Symmetric cryptosystems: Block Cipher (DES, Skipjack….), Asymmetric cryptosystems: RSA, ECC; Key Management and distribution; Digital signature, Hash functions, Smart Card security; Public Key Infrastructure: PKCS Standards, X. 509 Certificates, Certificate Policies and Cross Certification; Java Cryptography Architecture; Digital Signature and PEC in the Italian law.

Identification and Authentication mechanism: Authentication mechanisms, Authentication protocols, Single Sign On, Kerberos, Identity Federation, OAuth, SAML, IAM (Identity and Access Management) Systems; credential management systems (Vault).

Access Control mechanism: Access Control models: Discretionary and Mandatory Access Control Models (DAC, MAC), Role based Access Control Models (RBAC), Other models: Attribute based Access Control (ABAC), Role hierarchy management, Conflict management; Access Control frameworks: XACML, Keycloack, Authentication and Authorization services.

System and Communication Protection mechanism: Attack taxonomy, Firewalls, Gateways, Intrusion Detection systems; Network segmentation and demilitarized zone (DMZ), Monitoring mechanisms; Auditing and Logging mechanisms.
Application and Network Security protocols: SSL, PGP, SMIME, VPN, IPv6.

Design of Secure Systems: standard risk-based development approach (NIST, ISO), Secure SDL methodologies, threats and vulnerabilities analysis, risk analysis, security controls identification techniques, security assessment, static and

dynamic security testing techniques. Design trade-offs: Security and Performances. Case studies: Web application security, Security in hw and embedded devices (IoT security, WSN security, FPGA security…..), Cloud Security.

MATERIALE DIDATTICO

Libro di testo: Stallings William – Computer Security, Principles and Practice 3rd Ed - Prentice Hall.
Dispense e presentazioni fornite dal docente relative ad argomenti teorici e applicativi.
Manuali e standard di riferimento dei meccanismi e metodologie di sicurezza utilizzati.
Codice relativo alle esercitazioni svolte in aula.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso prevede circa il 70% di lezioni frontali in cui vengono affrontati gli argomenti teorici, mentre il restante 30% è riservato a lezioni pratiche ed esercitazioni riguardanti la progettazione, implementazione e valutazione di meccanismi di sicurezza studiati.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	x
discussione di elaborato progettuale	x
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

(*) È possibile rispondere a più opzioni

La verifica dell'apprendimento prevede una prova orale orientata alla verifica della comprensione dei concetti teorici del corso e alla discussione di un elaborato.

SCHEDA DELL'INSEGNAMENTO (SI)

"NETWORK SECURITY"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: PROF. SIMON PIETRO ROMANO
TELEFONO: +39 081 7683823
EMAIL: SIMONPIETRO.ROMANO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II):
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Scopo del corso è quello di fornire agli studenti nozioni avanzate nel campo della sicurezza di rete. Gli studenti acquisiranno familiarità con i più noti meccanismi di sicurezza, nonché con le tecniche di mitigazione degli attacchi informatici, concentrandosi sulle soluzioni disponibili ai vari livelli dello stack protocollare di rete, dallo strato fisico a quello applicativo.

Il corso fa leva su alcuni dei concetti trattati nel corso di "Secure Systems Design", con particolare riferimento alla crittografia simmetrica, alla confidenzialità dei messaggi, alla crittografia a chiave pubblica ed alla autenticazione. D'altro canto, esso introduce alcuni degli argomenti che costituiscono il nucleo del corso di "Software Security", quali, ad esempio, gli attacchi di tipo "buffer overflow" e le tecniche di fuzzing.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Alla fine del corso, gli studenti saranno in grado di dimostrare conoscenze avanzate nel campo della sicurezza di rete. Essi avranno un'idea chiara della catena vulnerabilità-minaccia-attacco e saranno capaci di progettare tecniche di difesa efficaci. Sapranno come proteggere una infrastruttura critica di rete. Saranno pronti ad entrare a far parte del mercato del lavoro come membri di team esperti in cybersecurity, contribuendo attivamente sia alle fasi di rilevamento degli attacchi di rete, che a quelle di reazione e di implementazione delle necessarie azioni correttive. Il completamento con successo del corso consentirà loro di portare sul campo le conoscenze maturate, giocando un ruolo da protagonisti all'interno dei cosiddetti Security Operation Center (SOC) e/o di team di professionisti specializzati in sicurezza informatica.

Conoscenza e capacità di comprensione

Gli studenti dovranno mostrare di saper comprendere ed approfondire le problematiche legate alla protezione efficace di una architettura di rete da attacchi informatici. Dovranno approfondire argomenti relativi alle relazioni che intercorrono tra la sicurezza a livello del singolo nodo, del Sistema Operativo, del software e della rete, inquadrando il tutto in una visione integrata che faccia uso di un approccio olistico. Il corso fornisce agli studenti conoscenze avanzate nell'ambito della cybersecurity, illustrando come far leva su strumenti sia metodologici che pratici, al fine di scoprire vulnerabilità, rilevare attacchi, analizzare paradigmi di interazione e modelli comportamentali degli utenti di una rete, progettare ed implementare misure di difesa adeguate, con un approccio di tipo sia "reattivo" che "proattivo". Tali strumenti consentiranno agli studenti di identificare le connessioni causali che intercorrono tra i concetti di vulnerabilità, minaccia ed attacco, nonché di comprendere le implicazioni derivanti dall'impiego di un approccio offensivo al miglioramento del livello complessivo di sicurezza di una infrastruttura di rete.

Capacità di applicare conoscenza e comprensione

Gli studenti dovranno mostrare capacità di prendere decisioni, valutandone le conseguenze, a partire dalle informazioni disponibili (traffico in tempo reale, tracce di traffico registrate, file di log, audit del registro di sistema, codice sorgente delle applicazioni, ecc.) al fine di gestire in maniera efficace le problematiche relative alla presenza di un possibile attacco distribuito in rete. Essi dovranno anche dimostrare una conoscenza avanzata delle tecniche e degli strumenti che possono essere impiegati al fine di prevenire un attacco, rilevarlo in tempo reale, mitigarne gli effetti e contrastarne la presenza tramite l'impiego di meccanismi di rimedio attivo. Il corso fornisce le competenze e gli strumenti necessari per applicare in scenari pratici le conoscenze acquisite, favorendo la capacità di impiegare un approccio metodologico che integri in maniera opportuna differenti tipi di contromisure (eventualmente disponibili a diversi livelli dello stack protocollare di rete), nell'ambito di un contesto omogeneo che consenta di prendere in considerazione le molteplici sfaccettature di un attacco alla sicurezza.

PROGRAMMA-SYLLABUS

Le principali proprietà di sicurezza di un sistema informatico saranno introdotte e discusse in dettaglio. Verranno poi presentati ed analizzati gli approcci per migliorare la sicurezza ai vari livelli dello stack protocollare di rete. Il corso adotta un approccio del tipo "offensive security". Saranno illustrati i concetti legati alla cosiddetta catena vulnerabilità-minaccia-

attacco. Verranno descritte le tecniche di preparazione di un attacco informatico, quali *footprinting*, *scanning* ed *enumeration*. La fase finale di un attacco, nota con il termine *exploitation*, verrà infine presentata. Saranno introdotti argomenti quali il firewalling, il rilevamento delle intrusioni, l'analisi del software malevolo e la protezione da attacchi di tipo DDoS (Distributed Denial of Service). Le principali tecniche di cosiddetto ethical hacking saranno infine presentate ed analizzate in dettaglio.

Syllabus:

- Network security: principi ed architettura
    - requisiti funzionali di sicurezza
    - minacce, attacchi, contromisure
- Sicurezza delle reti wireless
- Sicurezza al livello rete
    - la suite di protocolli IPsec
- Sicurezza al livello trasporto
    - Secure Socket Layer (SSL), Transport Layer Security (TLS), Secure Shell (SSH)
- Sicurezza al livello applicativo:
    - E-mail
    - Web
        - HTTPS
        - architettura di sicurezza di WebRTC (Web Real Time Communications)
- Cloud Computing e sicurezza (cenni)
- Introduzione al software malevolo
    - tassonomia
        - Virus, worm, Trojan, rootkit, ecc.
    - Advanced Persistent Threats (APTs)
    - contromisure
- Attacchi "Denial of Service" (DoS) e "Distributed Denial of Service" (DDoS)
- Intrusion Detection Systems (IDS)
    - tecniche "host-based", "network-based" e ibride
- Firewall ed Intrusion Prevention Systems (IPS)
- Hacking in reti IP
    - Fasi preliminari di un attacco:
        - Footprinting, scanning, enumeration
    - Tecniche di attacco orientate:
        - agli end-system ed ai server;
        - all'infrastruttura:
            - reti VoIP (Voice over IP)
            - reti Wireless
            - sistemi hardware
        - ai dati ed alle applicazioni:
            - Web
            - dispositivi mobili
            - Database

MATERIALE DIDATTICO

1. *"Network Security Essentials Applications and Standards", 6th Edition, William Stallings, Published by Pearson (July 13th, 2021) – ISBN-13: 9780134527338, Copyright © 2017*
2. *"Computer Security: Principles and Practice", 4thEdition, William Stallings and Lawrie Brown, ISBN-13:* 9780134794105, *©2018, Pearson*
3. *"Hacking Exposed", 7th Edition by Stuart McClure, Joel Scambray and George Kurtz Mc Graw Hill ISBN-10:*

0071780289, ISBN-13: 978-0071780285

4. *Riferimenti ufficiali quali, ad esempio, Requests For Comments (RFC), disponibili sul sito: https://www.ietf.org*
5. *Trasparenze ed ulteriore materiale di approfondimento reso disponibile dal docente sui siti ufficiali del corso.*

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà un approccio pratico e partecipativo per tutta la durata del corso. Gli argomenti del corso saranno presentati in teoria ed ulteriormente analizzati attraverso esempi pratici e dimostrazioni.
Uno strumento fondamentale per il corso sarà rappresentato dal Docker Security Playground (DSP – https://github.com/giper45/DockerSecurityPlayground), una architettura open source basata sull'impiego del paradigma a microservizi ed impiegata per la realizzazione di infrastrutture virtuali appositamente specializzate per lo studio della sicurezza di rete. Un ricco insieme di laboratori di sicurezza pre-configurati è reso disponibile dal docente sotto forma di repository github, raggiungibile all'URL https://github.com/NS-unina/DSP_Repo.
Tutti gli studenti dovranno lavorare, individualmente o in gruppo, ad un progetto pratico focalizzato sullo studio approfondito di uno o più degli argomenti del corso.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

b) Modalità di valutazione:

- 35% del voto finale dipenderà dalla valutazione del progetto pratico sviluppato dallo studente (individualmente o in gruppo). Tale progetto dovrà essere consegnato al docente (completo di documentazione e comprensivo, qualora presente, del codice sorgente sviluppato) almeno sette giorni prima della data fissata per la prova orale;
- 65% del voto finale dipenderà dall'esito della prova orale.

SCHEDA DELL'INSEGNAMENTO (SI)

"CYBERSECURITY DATA ANALYSIS"

SSD ING INF 05*

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ANTONIO PESCAPE'
TELEFONO: 0817683856
EMAIL: ANTONIO.PESCAPE@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): II
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nozioni di protocolli di rete Internet.

OBIETTIVI FORMATIVI

L'obiettivo dell'insegnamento è quello di fornire agli studenti le nozioni specialistiche utili all'analisi di una moderna rete internet con particolare riferimento agli aspetti legati alla sicurezza di rete. Il corso presenta i contenuti adottando un approccio ingegneristico ed empirico e fonde lezioni teoriche, lezioni pratiche, seminari ed esercitazioni. Esso presenta in modo approfondito gli aspetti principali e le motivazioni alla base dell'analisi di una rete di calcolatori, per poi approfondire gli aspetti metodologici e pratici legati all'analisi di rete con un focus specifico sulla analisi, identificazione e classificazione di eventi anomali quali, ad esempio, attacchi informatici. L'obiettivo è studiare le principali tecniche, tecnologie e strumenti per il monitoraggio e l'analisi del traffico di rete e le loro applicazioni alla cybersecurity. Il corso esamina i principali approcci per effettuare misurazioni della rete Internet (monitoraggio passivo e attivo), le tecniche utilizzate per identificare il traffico di applicazioni e servizi (classificazione del traffico) e i metodi utilizzati per la sicurezza di rete, come il rilevamento e la mitigazione degli attacchi, utilizzando approcci ad apprendimento automatico (Machine Learning). Il corso prevede inoltre attività sperimentali finalizzate alla redazione di una relazione tecnico-scientifica.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e saper comprendere le problematiche relative all'analisi e al monitoraggio delle reti Internet e del traffico di rete, sia benigno sia malevolo. Deve dimostrare di sapere elaborare argomentazioni concernenti le relazioni tra il traffico di rete e fenomeni quali attacchi, malfunzionamenti di rete, problematiche prestazionali. Tali strumenti consentiranno agli studenti di comprendere le connessioni causali tra l'uso di applicazioni di rete, il traffico generato da tali applicazioni e le condizioni di esercizio della rete in presenza sia di traffico benigno sia malevolo. Inoltre, ci si aspetta che lo studente sia in grado di riconoscere le relazioni tra le operazioni eseguite attraverso la rete (esecuzione di applicazioni, configurazione di dispositivi di rete, attacchi a dispositivi di rete) e gli eventi osservabili (mutamenti nelle caratteristiche del traffico e nelle funzionalità dei dispositivi di rete).

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di trarre le conseguenze da un insieme di informazioni per identificare e risolvere problemi concernenti le infrastrutture e le applicazioni di rete, pianificare infrastrutture e servizi sulla base dei requisiti di sicurezza richiesti alle comunicazioni effettuate tramite la rete Internet. Analizzando i dati di rete, lo studente deve essere in grado di inferire la natura (benigna/malevola) e caratteristiche volumetriche della comunicazione (numero di dispositivi coinvolti, quantità e tempistica dei messaggi scambiati). Lo studente deve essere in grado di applicare gli strumenti metodologici appresi ai seguenti ambiti: raccolta dei dati di traffico Internet, analisi delle applicazioni di rete e sicurezza di rete. Più precisamente, le analisi di sicurezza eseguite osservando e analizzando scenari operativi di rete, costituiscono un approccio proattivo alla sicurezza informatica, che utilizza capacità di raccolta, aggregazione ed analisi dei dati al fine di portare a termine funzioni di sicurezza essenziali che rilevano, analizzano e mitigano le minacce informatiche alle reti.

PROGRAMMA-SYLLABUS

Introduzione, Concetti di Base e Fondamenti: Contestualizzazione didattico/scientifica del Corso, Terminologia di Base, Inquadramento degli aspetti principali dell'analisi di Internet e motivazioni (precedenti e successivi al proliferare di attacchi informatici), Risoluzione dei Problemi di Rete, Requisiti delle Reti, Sicurezza di Rete, Analisi di Sicurezza; Background Analitico (probabilità, statistica, rappresentazione dei dati, forecasting, grafi, etc.), Metodologie e Tecniche di Machine/Deep Learning e Data Mining; Task di Apprendimento: Classificazione, Predizione e Anomaly Detection; Tecniche di eXplainable AI (XAI) per approcci Machine/Deep Learning; Reinforcement Learning; Adversarial Learning; Continuous e Few Shot Learning; Framework di Valutazione e Metriche di Prestazione; Data Visualization. [1,5 CFU]

Dati e Monitoraggio di Rete: Modelli e Metriche per l'Analisi ed il Monitoraggio di Internet; Approcci al Monitoraggio di Internet (attivo, passivo, ibrido, etc.); Dalla Rete al Traffico: Metodologie e strumenti per l'acquisizione e la caratterizzazione di dataset di traffico di rete (generati da utenti e generati da bot); Tipi di dato per la sicurezza della rete (pacchetti di rete, feature e statistiche estratte, log file di rete e di sistema); Dataset pubblici per l'analisi del traffico di

rete e la sicurezza di rete: caratterizzazione e principali utilizzi; Gestione e configurazione degli strumenti per il monitoraggio di rete; Metodologie e Tecniche per l'analisi del Traffico di Internet: workload di rete, caratterizzazione e modelling statistico, Traffico Self-Similare, Modelli di generazione del Traffico Internet; "Practical Issues" nell'Analisi e nel Monitoraggio di Internet: middleboxes (PEP, PDP, Firewall, etc). [1,5 CFU]

Analisi dei Dati e Sicurezza Informatica: Dal traffico ai dati, dai dati alle feature: analisi ed interpretazione dei dati di rete, analisi e selezione delle feature di rete; Identificazione e Classificazione del traffico di rete con approcci basati sull'intelligenza artificiale; Identificazione e Classificazione del traffico di rete anomalo e malevolo (malware, attacchi DoS/DDoS, BotNet, ecc.) con particolare riferimento agli scenari 5G, IoT (Industriale), Cloud, Web e Mobile; Identificazione delle Anomalie Automatica, Zero-day Detection, Machine Learning Network Intrusion Detection Systems (ML-NIDS), nuovi attacchi a sistemi ML e "adversarial input perturbations", Piattaforme sperimentali su larga scala per l'analisi e il monitoraggio del traffico Internet e generato da attacchi; Neutralità di Rete ed applicazione, rilevamento ed elusione di meccanismi di censura; Robustezza agli attacchi ai percorsi di rete ed alle topologie (ad esempio, attacchi al protocollo di instradamento); Sfide legali ed etiche che emergono dalla raccolta di dati sulle attività di utenti umani e dal loro utilizzo al fine di costruire modelli di apprendimento automatico. [3 CFU]

MATERIALE DIDATTICO

Il materiale didattico è costituito dalle Slide, dalle Dispense e dagli articoli forniti dal docente.
Libro di testo di approfondimento: *Internet Measurement: Infrastructure, traffic & applications, Mark Crovella, Balachander Krishnamurthy, Wiley.*

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà:
a) lezioni frontali per circa il 40% delle ore totali,
b) esercitazioni per approfondire praticamente aspetti teorici per circa il 40% delle ore totali,
c) seminari per approfondire tematiche specifiche per circa il 20% delle ore totali.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità d'esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

b) Modalità di valutazione:

SCHEDA DELL'INSEGNAMENTO (SI)

SOFTWARE SECURITY

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ROBERTO NATELLA
TELEFONO: 081 7683820
EMAIL: ROBERTO.NATELLA@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): II
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Sono sufficienti come prerequisiti le nozioni di base acquisite nel corso della Laurea Triennale in Ingegneria Informatica sui sistemi operativi, l'ingegneria del software, le basi di dati, le architetture dei calcolatori elettronici, e le reti di calcolatori.

OBIETTIVI FORMATIVI

L'obiettivo generale dell'insegnamento è di fornire agli studenti nozioni e competenze per la progettazione, lo sviluppo, la validazione e la gestione di sistemi software sicuri. Questo obiettivo è complementare altri insegnamenti della area di "Cyber-Security" della Laurea Magistrale in Ingegneria Informatica.

Gli obiettivi formativi dell'insegnamento includono:

- Fornire concetti avanzati sulle minacce, sulle vulnerabilità, e sugli attacchi ai sistemi software.
- Fornire competenze specialistiche di progettazione e sviluppo sicuro del software.
- Fornire competenze specialistiche per la validazione del software mediante analisi statica e dinamica.
- Fornire concetti avanzati sulle tecniche adottate nel software malevolo.
- Fornire competenze specialistiche per la prevenzione e la rilevazione di attacchi dovuti a software malevolo.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e saper comprendere le minacce, le vulnerabilità, e gli attacchi nei sistemi software, e le tecniche utilizzate nel software malevolo. Il percorso formativo intende fornire agli studenti le conoscenze e gli strumenti metodologici per comprendere le attuali e future problematiche di sicurezza dei sistemi software a fronte della loro continua evoluzione nel tempo, e per identificare efficacemente le problematiche di sicurezza e comunicarle agli stakeholder coinvolti nella realizzazione e gestione dei sistemi software sicuri.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di realizzare nuovi sistemi software sicuri, applicando le capacità e gli strumenti metodologici e operativi acquisiti nel percorso formativo nel contesto di tutte le fasi del processo di sviluppo software (analisi, progettazione, sviluppo, testing, gestione). Il percorso formativo è orientato a favorire la capacità di effettuare scelte di progetto dei sistemi software in maniera consapevole delle problematiche di sicurezza, e di saper riconoscere e risolvere le minacce, le vulnerabilità, gli attacchi, e le tecniche di software malevolo all'interno di un sistema software.

PROGRAMMA-SYLLABUS

Introduzione alla software security: Il ruolo delle vulnerabilità software nelle minacce di sicurezza, enumerazione delle vulnerabilità (CVE, CVSS, CWE, CPE), top design flaws & bugs, disclosure e gestione delle vulnerabilità.

Vulnerabilità software: Buffer overflow, vulnerabilità di memory & type safety (format strings, use-after-free, double free, out-of-bounds reads, undefined behaviors), web vulnerabilities (session hijacking, CSRF, XSS, SQL/OS command injection, path traversal, SSRF).

Progettazione sicura del software e certificazione: Processi e standard di software security (Microsoft Security Development Lifecycle, Common Criteria, OWASP, CERT), principi di progettazione sicura del software, threat modeling, security requirements e abuse cases, software supply chain security, DevSecOps.

Programmazione software sicura: Tecniche di difesa OS/compiler/language/framework-based, strategie di input validation, espressioni regolari, tecniche di error handling e resource management.

Tecniche di identificazione delle vulnerabilità: Analisi statica del software (type checking, compiler-based checks, quality scanners, security scanners), analisi dinamica del software (source/binary code instrumentation, sanitizers), fuzzing (generation/mutation fuzzing, coverage-driven fuzzing, library-based fuzzing, protocol fuzzing).

Software malevolo: Forme di software malevolo (RAT, keyloggers, rootkits, etc.), Advanced Persistent Threats (APTs), Cyber Threat Intelligence, modellazione degli attaccanti (Cyber Kill Chain, Diamond Model, MITRE ATT&CK), tecniche di attacco nel sistema operativo Windows (living-off-the-land malware, process injection, furto di credenziali, persistenza, abuso di privilegi, e altre), reverse engineering di codice binario, rilevazione del malware mediante firme (Snort, Yara, Sigma).

MATERIALE DIDATTICO

Libri di testo:

- Wenliang Du, ***"Computer & Internet Security: A Hands-on Approach"***, 2a edizione, 2019
- A. Takanen, J. DeMott, C. Miller, A. Kettunen, ***"Fuzzing for Software Security Testing and Quality Assurance"***, 2a edizione, 2018
- B. Chess, J. West, ***"Secure Programming with Static Analysis"***, 1a edizione, 2007
- M. Sikorski, A. Honig, ***"Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software"***, 1a edizione, 2012

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Le attività dell'insegnamento saranno strutturate in:

- Lezioni frontali, per circa il 60% delle ore totali
- Attività di laboratorio per applicare e approfondire le conoscenze acquisite, per circa il 40% delle ore totali

Le lezioni si avvarranno di dimostrazioni pratiche basate su ambienti di virtualizzazione, e sugli strumenti di sicurezza software maggiormente utilizzati nei sistemi operativi Linux e Windows.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	
discussione di elaborato progettuale	X
altro	

b) Modalità di valutazione:

La valutazione avviene tramite la discussione di un elaborato progettuale, che lo studente sviluppa in autonomia da solo o in collaborazione con un piccolo gruppo di altri studenti. La discussione mira ad accertare la acquisita conoscenza da parte dello studente delle vulnerabilità e attacchi nei sistemi software, e la sua capacità di applicare operativamente tali conoscenze nella progettazione e/o nella validazione della sicurezza di un sistema software. La valutazione finale tiene in considerazione la complessità del progetto affrontato, la qualità del software eventualmente sviluppato e/o il grado di approfondimento nella validazione della sicurezza di un sistema software, e il grado di maturità dimostrato nella esposizione delle problematiche di sicurezza.

SCHEDA DELL'INSEGNAMENTO (SI)

"SOFTWARE ARCHITECTURE DESIGN"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI – DOCENTE

DOCENTE: ANNA RITA FASOLINO
TELEFONO: 081 7683906
EMAIL: ANNARITA.FASOLINO@UNINA.IT

INFORMAZIONI GENERALI – ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III):
SEMESTRE (I, II): I
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Ingegneria del Software

OBIETTIVI FORMATIVI

L'insegnamento si propone di introdurre il tema delle architetture software e di fornire agli studenti conoscenze e competenze specialistiche relative alla progettazione, documentazione e sviluppo di tali architetture.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e comprendere le problematiche relative alla disciplina delle architetture software ed il loro ruolo nell'ambito dei moderni processi di sviluppo software; il percorso formativo intende fornire agli studenti le conoscenze necessarie per comprendere le caratteristiche dei diversi modelli, stili e pattern architetturali e gli strumenti metodologici necessari per progettare, documentare, e sviluppare una architettura software. Tali conoscenze e strumenti consentiranno agli studenti di comprendere le principali relazioni che sussistono fra progetto architetturale e conseguenti caratteristiche di qualità del software e di cogliere le implicazioni delle diverse scelte di progetto sui requisiti posseduti dall'architettura risultante.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di saper utilizzare le conoscenze e gli strumenti metodologici appresi per riuscire a progettare, documentare ed implementare architetture software in grado di realizzare tutti i requisiti funzionali e di qualità richiesti, implementando un processo di sviluppo ispirato ai moderni processi iterativi, evolutivi ed agili.

PROGRAMMA-SYLLABUS

Architetture Software: concetti fondamentali.
Definizioni di Architettura Software. Architettura come insieme di Strutture e di Viste: Strutture Modulari, Strutture dei Componenti e Connettori, Strutture di Allocazione. Ruolo ed Importanza dell'architettura nei processi di sviluppo software.

Qualità delle Architetture Software.
Attributi di qualità di una architettura osservabili a run-time (performance, security, availability, functionality, usability), Attributi non osservabili a run-time (modifiability, portability, reusability, integrability, e testability), attributi di qualità intrinseca dell'architettura (conceptual integrity, correctness, and completeness, buildability).
Specificare gli attributi di qualità attraverso la tecnica degli Scenari di qualità. (CENNI)

Progettazione Architetturale.
Principi di progettazione architetturale, Progettazione basata sul riuso di esperienza, attraverso Pattern, Stili e Tattiche Architetturali.

Pattern Architetturali.
I Pattern Architetturali State-logic-Display, Sense-Compute-Control e Model-View-Controller (MVC). Esempi.

Stili Architetturali
Stile procedurale e object-oriented, Stile a Livelli (Macchine Virtuali e Client-Server), Data-Flow (Batch e Pipe-and-Filter), a Memoria Condivisa (Blackboard e Rule Based), basato su Interprete (Codice Mobile), ad Invocazione Implicita (Publish-Subscribe e Event-Based), Stile ad Oggetti Distribuiti, Stile Service Oriented (SOA), Stile a Microservizi.

Tattiche per realizzare gli attributi di qualità di una architettura
Tattiche per l'Availability, Deployability, Modifiability, Energy Efficiency, Performance, Security, Safety, Testability, Usability.

Soluzioni Architetturali
Accesso alle risorse di una architettura attraverso le Interfacce; progettazione delle Interfacce e Stili di interazione ; la rappresentazione e la struttura delle risorse scambiate (XML, JSON e Binary); Virtualizzazione ed il Cloud Computing; le applicazioni Mobili

Documentare le Architetture Software.
Documentare le diverse viste di una architettura software: stili di documentazione per la vista modulare, dei componenti e connettori, per la vista di distribuzione. Documentare il Contesto, il comportamento, le decisioni di progetto e le interfacce dei componenti di una architettura. Cenni allo Standard ISO/IEC 42010 (Systems and Software Engineering- Architecture Description).

Sviluppo di architetture Software con processi Iterativi ed Incrementali.
Lo sviluppo iterativo ed Incrementale: differenze rispetto allo sviluppo waterfall. Unified Process (UP): le Fasi di Ideazione, Elaborazione, Costruzione e Transizione. Le discipline e le pratiche di UP.
Sviluppo Agile. I Principi dello Sviluppo Agile. Manifesto Agile e pratiche Agili. Il Test Driven Development (TDD) e il Refactoring. Continuous Integration (CI). Il framework SCRUM: i ruoli in Scrum, i Backlog, le pratiche di Scrum.

Sviluppo Iterativo ed i Processi di Gestione della Configurazione del Software
La gestione delle Versioni del software. Modello di Version Control Centralizzato e Distribuito. Il processo di Build e l'Agile Building, Continuous Integration, DevOps. I processi di gestione delle Richieste di cambiamento del Software. Gestione delle Release.

MATERIALE DIDATTICO

- Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, Fourth Edition, Addison Wesley, 2022
- P. Clements, F. Bachmann, Bass, etc. Documenting Software Architectures- Views and Beyond, Addison Wesley, Second Ed., 2010.
- N. Taylor, N. Medvidovic, E. Dashofy. Software Architecture -foundations, theory and practice, Wiley 2010.

Copia delle Slide proiettate durante le lezioni
Articoli scientifici e materiale integrativo dal Web

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

L'insegnamento verrà erogato prevalentemente attraverso lezioni frontali e sarà integrato da esercitazioni pratiche e da seminari di approfondimento. Le esercitazioni verteranno sullo studio di architetture software reali e l'applicazione dei metodi di progettazione e documentazione delle architetture presentati a lezione. Lo svolgimento delle esercitazioni potrà richiedere l'uso di ambienti integrati e software specialistico per la progettazione, la documentazione ed implementazione di architetture software.

Il docente utilizzerà:
a) lezioni frontali per 30 ore (pari a circa il 60% delle ore totali),
b) esercitazioni per approfondire praticamente aspetti teorici e applicare le conoscenze apprese per 14 ore (pari a circa il 30% delle ore totali),
c) seminari specialistici per 4 ore

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità d'esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
Solo orale	X
discussione di elaborato progettuale	
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

La prova orale finale tende ad accertare le conoscenze apprese nel campo delle architetture software, nonché dei moderni metodi e processi di progettazione e sviluppo di architetture software. La prova orale potrà prendere spunto da un elaborato di approfondimento svolto dagli studenti su uno dei temi trattati a lezione.

b) Modalità di valutazione:

SCHEDA DELL'INSEGNAMENTO (SI)

"SOFTWARE TESTING"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: PORFIRIO TRAMONTANA
TELEFONO: 0817683901
EMAIL: PORFIRIO.TRAMONTANA@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III):II
SEMESTRE (I, II): I
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

I prerequisiti fondamentali di questo corso sono la conoscenza di base della programmazione ad oggetti, ed in particolare del linguaggio Java. Inoltre, è necessaria una conoscenza di base delle problematiche relative all'Ingegneria del Software. Nessuna conoscenza preliminare relativa al testing è invece necessaria.

OBIETTIVI FORMATIVI

Il corso si propone di definire e approfondire tematiche di Verifica e Validazione del Software, includenti metodologie, strategie, tecniche, strumenti e processi di Software Testing e Debugging. Il corso si propone di contribuire all'acquisizione di competenze relative allo sviluppo di software di qualità e alla sua valutazione, con particolare attenzione all'automazione di tali attività.

Questi obiettivi sono di fondamentale importanza nella realizzazione e valutazione di qualsiasi impianto informatico e sistema di elaborazione, da quelli hardware a quelli software, dai sistemi operativi alle reti di elaboratori, dalle basi di dati ai sistemi informativi, dall'intelligenza artificiale alla robotica.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve essere in grado di comprendere l'importanza della qualità del software e della sua valutazione, in particolare la ricerca e correzione dei suoi difetti tramite l'applicazione di adeguate strategie e tecniche di testing. Lo studente deve comprendere quali siano le difficoltà intrinseche nel testing di un sistema software e saper scegliere tra le soluzioni che possono essere applicate. Tali conoscenze permetteranno agli studenti di poter integrare le conoscenze relative all'ingegneria del software allo scopo di organizzare processi di sviluppo di software di qualità e ad alta affidabilità. Lo studente deve anche essere in grado di comprendere come le metodologie e tecniche apprese possano contribuire a risolvere problemi relativi alla realizzazione di sistemi hardware o software di qualità.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di saper organizzare processi di sviluppo software che prevedano una continua valutazione e miglioramento della qualità grazie all'esecuzione di efficaci ed efficienti attività di testing. Lo studente deve essere in grado di progettare e sviluppare piani di test del software a diversi livelli di dettaglio, dal testing di unità fino al testing di sistema e a quello di accettazione. Lo studente deve essere in grado di realizzare casi di test con elevato livello di automazione, sia in fase di generazione dei casi di test, che in fase di esecuzione e valutazione dell'esito. Lo studente deve essere in grado di utilizzare al meglio gli strumenti di testing esistenti e di aver acquisito la capacità di progettare e realizzare strumenti di testing innovativi.

PROGRAMMA-SYLLABUS

Elementi teorici del testing software: Definizioni – Problemi indecidibili – Tassonomia delle attività di testing

Qualità del testing: Adeguatezza - Precisione - Ripetibilità - Capacità di trovare i difetti - Efficacia – Efficienza

Specifica dei casi di test: Input – Output – Oracoli – Pre e post-condizioni

JUnit: Introduzione a JUnit - Implementazione di test di unità con JUnit su programmi Java – Assunzioni e asserzioni – Testing delle eccezioni – Test dinamici e parametrici – Testing Data Driven con JUnit

Testing Black Box. Testing basato sui requisiti e sugli scenari dei casi d'uso - Test con classi di equivalenza e valori limite - Strumenti e tecniche per la generazione combinatoriale dei casi di test - Testing con tabelle di decisione.

Testing White Box – Modelli e metriche di copertura - Strumenti per la misura automatica della copertura del codice.

Testing di integrazione e testing in isolamento – Tecniche di testing in isolamento con driver e stub – Grafi delle dipendenze - Strategie per il testing di integrazione: top-down, bottom-up, sandwich – Testing con Mock Objects – Cenni di dependency injection – Framework per la creazione di Mock Objects

Testing dell'interfaccia utente – Tecniche di testing delle interfacce utente a caratteri - Testing delle GUI - Modellazione delle GUI con macchine a stati finiti - Problema dell'esplosione degli stati e tecnica degli stati equivalenti – Librerie a supporto del testing di GUI - Validazione degli input.

Tecniche User Session: Tassonomia delle tecniche Capture & Replay per la generazione di test sull'interfaccia utente - Problematiche relative alla generazione di locatori robusti – Problematiche legate alla replicabilità dei test catturati – Strumenti di Capture & Replay per applicazioni Web

Tecniche di testing automation: Automazione nella generazione/esecuzione/valutazione dell'esito dei casi di test - Generazione e valutazione automatica di oracoli - Crash testing - Smoke Testing - Regression Testing.

Random testing: Caratteristiche e parametri del random testing - Problema della terminazione del random testing - Tecniche e strumenti per l'esplorazione automatica della GUI - Riduzione e prioritizzazione delle test suite.

Mutation Testing: Test case mutation – Mutation Analysis – Mutation based Testing – Strumenti per la generazione di mutant - Search based software testing: utilizzo di EvoSuite.

Experience based Testing: Exploratory Testing – Error Guessing e Checklist based Testing – Beta Testing - Crowdtesting – Piattaforme di CrowdTesting - Software Testing Gamification – Testing di unità con Code Defenders

Testing in Continuous Integration: Cenni su tecniche, linguaggi e strumenti per la build automation - Cenni su tecniche e strumenti per la gestione delle versioni concorrenti – Automazione di attività di testing in Github con Github Actions

Analisi statica del codice sorgente: Tecniche per l'analisi statica - Strumenti automatici di analisi statica: Checkstyle, PMD, Findbugs, Android Lint - Cenno all'utilizzo di SonarQube per il monitoraggio continuo della qualità del software

Debugging: Localizzazione e correzione dei difetti - Tecniche per il debugging: forza bruta - backtracking, eliminazione delle cause - Strumenti per il debugging: breakpoint, breakpoint condizionali, watch, watchpoint.

Testing delle applicazioni Android. Introduzione al sistema Android e alla programmazione di app Android - Testing di unità: utilizzo di JUnit e Robolectric - Testing della GUI: utilizzo di Robotium e Android Espresso - Utilizzo di Espresso Recorder per il Capture & Replay di casi di test - Testing di sistema: utilizzo di UIAutomator - Strumenti di testing a basso livello: Monkey - Strumenti di monitoraggio - Testing dei memory leaks - Strumenti di testing sistematico: Android Ripper - Testing di applicazioni con risorse remote: Cloud testing con Firebase TestLab, Alpha Testing, Beta Testing.

MATERIALE DIDATTICO

Il materiale didattico comprende un insieme completo di presentazioni realizzate dal docente che accompagnano tutte le lezioni del corso. In queste presentazioni sono riportati riferimenti a numerosi libri, articoli scientifici, strumenti di sviluppo, materiale sperimentale, esempi ed altre risorse online, che sono tutte messe a disposizione degli studenti.

Tra i libri consigliati:

- Ian Sommerville, Ingegneria del Software, Pearson Addison Wesley
- Mauro Pezzè and Michal Young, Software Testing and Analysis, John Wiley & Sons, 2008

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso si compone di 48 ore di lezione complessive, nelle quali si alterna didattica frontale (36 ore) con presentazione e discussione degli argomenti del corso ed esercitazioni (12 ore), nelle quali vengono mostrati e provati interattivamente tecniche e strumenti software a supporto delle attività di testing. Per alcuni argomenti sono proposte delle sperimentazioni basate sulla realizzazione di casi di test per sistemi software con l'applicazione di tecniche innovative di testing.
Vengono fornite le registrazioni di tutte le lezioni (prese in diretta mentre si tengono), il codice di tutti gli esempi ed esercitazioni svolte. Per le attività pratiche sono utilizzati esclusivamente software aperti e di libero utilizzo e vengono fornite le indicazioni fondamentali sul loro reperimento e utilizzo.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
Altro	

L'esame si articola in uno o due elaborati progettuali, seguiti da una prova orale. Uno dei due elaborati progettuali consiste nella progettazione e realizzazione di un piano di test su di un software specifico, eventualmente proposto dal docente, applicando le tecniche e strumenti presentati al corso. Il secondo elaborato, invece, consiste nell'approfondimento di una tematica di testing innovativa rispetto ai contenuti del corso o nella realizzazione di uno strumento innovativo di testing del software.
La consegna e discussione con esito positivo degli elaborati di natura progettuale consente di accedere alla prova orale. La prova orale consiste di una valutazione complessiva del raggiungimento degli obiettivi di apprendimento previsti, tramite interrogazione su argomenti relativi a tutto il programma del corso.
Il voto finale terrà in considerazione sia della correttezza e innovatività degli elaborati progettuali, che della valutazione della prova orale.

SCHEDA DELL'INSEGNAMENTO (SI)

"COGNITIVE COMPUTING SYSTEMS"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: PAOLO MARESCA
TELEFONO: +39 081 7683168
EMAIL: PAOLO.MARESCA@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): II
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Obiettivo dell'insegnamento è quello di fornire le conoscenze e competenze approfondite necessarie per la comprensione di sistemi basati sul paradigma del cognitive computing. Il cognitive computing è una disciplina emergente che, mettendo insieme conoscenze di cloud, Big Data, IOT, connessioni fra le reti, machine learning, natural language processing, AI, deep learning e knowledge representation, sviluppa sistemi automatici che cercano di simulare il processo del pensiero umano. Gli studenti avranno anche l'opportunità di maturare le competenze specialistiche necessarie per lo sviluppo di applicazioni cognitive che possono interagire con persone e/o cose (macchine e/o altri computer). Il corso sarà corredato da una attività di esercitazione e sviluppo di applicazioni in laboratorio.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve essere in grado di conoscere le tematiche e le problematiche ricomprense in questo corso fino al punto di elaborare connessioni trasversali fra le discipline che il cognitive computing ricomprende. Il paradigma del cognitive computing è dirompente (disruptive) perché introduce uno sconvolgimento nel modo di procedere nel risolvere problemi e nel progettare soluzioni di fronte a problemi che spaziano in molte tematiche e su molti domini applicativi. L'insegnamento fornirà gli strumenti metodologici per elaborare ed applicare questo nuovo paradigma su casi concreti.

Capacità di applicare conoscenza e comprensione

Lo studente sarà in grado, da solo o in gruppo, di ideare, progettare e realizzare una applicazione cognitiva. Egli acquisirà gli strumenti metodologici propri del paradigma del cognitive computing e li applicherà al dominio applicativo designato. La sua capacità nel dominare la pervasività delle applicazioni cognitive sarà importante per applicare concretamente le conoscenze maturate nel corso. Allo scopo di implementare correttamente una soluzione, esperienze provenienti dalle aziende più quotate in ambito mondiale faranno da traccia nella maturazione degli strumenti metodologici all'avanguardia nelle aziende e nei centri di ricerca.

PROGRAMMA-SYLLABUS

Il programma sarà composto dai seguenti argomenti

- An overview of Artificial Intelligence and Machine Learning
- Natural Language Understanding
- Approaches to automated question answering and relation extraction
- The foundation of cognitive computing
- Design principles of cognitive computing and Design Thinking
- Cognitive computing ecosystem scenario
- Applying Advanced Analytics to cognitive computing
- Unstructured Information Management Architecture & Massively Parallel Probabilistic Evidence-Based architecture
- The Cognitive computing paradigm
- Cooperative cognitive programming & Flow-Based programming
- The role of cloud and distribute computing in CC: PaaS platform as IBM-Watson, Microsoft-Azure, et al
- Building a Watson-enabled system
- The process of Building a cognitive application
- Applications of cognitive computing to Internet of Things
- The operation of AI platforms
- Efficient use of resources
- Responsible AI
- Approach that combines small data & wide data
- New trends in Cognitive computing

An overview of artificial intelligence and machine learning

In questo capitolo si fornisce una overview di intelligenza artificiale e di machine learning percorrendo la sua traccia nella ricerca negli ultimi 50 anni. La visione attuale di IA è quella di Intelligenza Aumentata. In altri termini l'IA non dovrebbe tentare di sostituire gli esseri umani, ma piuttosto assistere e amplificare la loro intelligenza. Possiamo usare l'intelligenza aumentata per estendere le capacità umane e realizzare cose che né gli uomini né le macchine potrebbero fare da soli. Alcune delle sfide che affrontiamo oggi provengono da un eccesso di informazioni. Internet ha portato a comunicazioni più veloci e all'accesso a una grande quantità di informazioni. Il calcolo distribuito e l'IoT hanno portato alla generazione di enormi quantità di dati e il social networking ha fatto si che la maggior parte di quei dati non siano strutturati. Ci sono così tanti dati che gli esperti umani non possono tenere il passo con tutti i cambiamenti e i progressi nei loro campi di studio: insomma non abbiamo un piano per utilizzarli! Con l'intelligenza aumentata o allargata (Augmented Intelligence), vogliamo mettere le informazioni di cui gli esperti hanno bisogno a portata di mano e restituire tali informazioni con una probabilità di correttezza, in modo che gli esperti possano prendere decisioni più informate. Vogliamo che gli esperti aumentino le loro capacità in modo che possano migliorare il servizio ai propri clienti. Vogliamo lasciare che le macchine facciano il lavoro dispendioso in termini di tempo in modo che gli esperti siano in grado di fare le cose che contano. Una delle sfide importanti è che l'intelligenza artificiale, sostituendo gli uomini nelle attività ripetitive di tutti i giorni, dovrà essere accettata in moltissimi campi della vita e del lavoro delle persone. I dati che vengono raccolti riguardano persone, elettrodomestici, automobili, reti, assistenza sanitaria, decisioni di acquisto, meteo, comunicazioni e tutto ciò che vediamo. È anche più facile che mai avere accesso a questi dati grazie agli smart phone e gli smart watch, gli utenti li usano per pubblicare immagini dei loro prodotti alimentari, acquisti di negozi ed esercizi di routine sui social media. Quando sviluppiamo sistemi di IA, dobbiamo stabilire fiducia. Si pone subito un problema che è : se non ti fidi di una macchina che fornisce delle raccomandazioni non puoi fidarti dei consigli che ti dà. Se qualcuno ti dà consigli, dobbiamo avere un livello di fiducia costruito prima di seguirli. Un modo per creare fiducia è mostrare come è stata costruita la raccomandazione (trasparenza). Qualsiasi soluzione che utilizza l'intelligenza artificiale dovrebbe essere costruita tenendo conto della trasparenza e della divulgazione. La privacy è un'altra preoccupazione che deve essere affrontata fin dall'inizio quando si costruisce una soluzione di intelligenza artificiale. Una volta persa la privacy, non puoi recuperarla. Poi ci sono i problemi etici che l'uso della IA crea Le sfide etiche ci sono sempre state nel momento in cui la tecnologia ha fatto passi avanti: si pensi ai primi anni della televisione etc., non ci meraviglia che questo accada oggi con l'IA. Semplicemente dobbiamo accettare queste sfide. Insomma un mondo stimolante e alla avanguardia ci aspetta in questo corso.

Natural language Understanding

Il linguaggio naturale è il modo con il quale gli esseri umani esprimono concetti, esigenze, sentimenti, etc. Esso rappresenta lo strumento di comunicazione principale. Sembra naturale ed ovvio iniziare da questo potente strumento per iniziare un viaggio attraverso lo studio e l'uso di questo per istruire macchine affinche facciano ciò che noi desideriamo esse facciano. Ma il linguaggio naturale o parlato ha molte sfumature: dialetti, toni, abbreviazioni, metafore, modi di dire, etc. I nativi della lingua parlata non hanno problemi a parlare e a scrivere tutto ciò ma per le macchine è tutta un altra storia. La PNL è uno dei sottocampi più importanti dell'apprendimento automatico per una serie di ragioni. Il linguaggio naturale è l'interfaccia più naturale tra un utente e una macchina. Una trattazione rigorosa porterebbe via un intero corso, ci basterà capire quali sono i punti interessanti della ricerca e come il linguaggio sia importante nel progettare sistemi cognitivi che interagiscono con l'utente e siano in grado di risolvere problemi di complessità sempre più crescente. Vedremo come questa complessità avrà un impatto sulle architetture hardware e software che supportano le applicazioni.

Approches to automated question answering and relation extraction

Nel 2007, IBM Research ha affrontato la grande sfida di costruire un sistema informatico in grado di competere con i campioni al gioco di Jeopardy !, Un quiz televisivo televisivo statunitense. Nel 2011, il sistema di risposta alle domande a dominio aperto, denominato Watson, ha battuto i due giocatori con il punteggio più alto in un incontro di due partite Jeopardy!

I progressi nella tecnologia di risposta alle domande (QA) aperte possono aiutare i professionisti a prendere decisioni critiche e tempestive in aree quali: conformità agli standard, sanità, integrità aziendale, business intelligence, knowledge discovery, gestione della conoscenza aziendale, sicurezza e assistenza clienti, ect.

Per realizzare un progetto di tale complessità serve una architettura adeguata. l'architettura del sistema di risposta alle domande (QA) progettata per consentire a Watson-IBM di giocare a Jeopardy! come DeepQA. DeepQA è un'architettura software per l'analisi approfondita dei contenuti e il ragionamento basato sull'evidenza. Rappresenta una potente capacità che utilizza l'elaborazione avanzata del linguaggio naturale (PNL), il recupero delle informazioni, il ragionamento e l'apprendimento automatico. La filosofia alla base dell'approccio di ricerca che ha portato a DeepQA è che la vera intelligenza emergerà dallo sviluppo e dall'integrazione di molti algoritmi diversi, ciascuno guardando i dati da diverse prospettive. Il successo del sistema di risposta alle domande di Watson può essere attribuito all'integrazione di una varietà di tecnologie di intelligenza artificiale.

L'architettura DeepQA vede il problema della risposta automatica alle domande come un compito di generazione e

valutazione di ipotesi massicciamente parallele. DeepQA può essere visto come un sistema che genera un'ampia gamma di possibilità e, per ciascuna, sviluppa un livello di fiducia raccogliendo, analizzando e valutando le prove basate sui dati disponibili.
Il principio computazionale primario supportato dall'architettura DeepQA può essere riassunto nei seguenti punti:

1. Assumere e perseguire molteplici interpretazioni della domanda.
2. Generare molte risposte o ipotesi plausibili.
3. Raccogliere e valutare molti percorsi di prove concorrenti che potrebbero supportare o confutare tali ipotesi.

The foundation of cognitive computing computing
Il capitolo affronta il paradigma del cognitive computing e perchè esso è differente da quello deterministico da cui discendono tutti i linguaggi di programmazione che lo studente conosce. Un paradigma basato sui dati piuttosto che sugli algoritmi e che si alimenta con una quantità sempre più crescente e mutevole di dati che esigono sistemi di calcolo paralleli e sempre più performanti. I dati cambiano velocemente. Ma perché questi dati crescono così velocemente ? perchè IL NOSTRO MODELLO SOCIALE E CAMBIATO. Le parole-chiavi che descrivono bene il modello di società in cui viviamo sono: mobile, sociotecnico, complesso e iperconnesso.
Una tecnologia cognitiva che si basa sul modo di ragionare dell'essere umano, in particolare quando un essere vivente osserva fenomeni e prende delle decisioni segue un approccio costituito da 4 passi. Ad esempio quando noi ci troviamo ad osservare qualche cosa e siamo chiamati a prendere una decisione (es. se salire le scale o prendere l'ascensore), il processo mentale che seguiamo è il seguente

1) osserviamo i fenomeni visibili e raccogliamo le prove oggettive,
2) usiamo quello che conosciamo per interpretare ciò che osserviamo generando ipotesi su ciò che viene osservato
3) valutiamo quali ipotesi sono giuste e quali sono sbagliate,
4) prendiamo la decisione scegliendo l'ipotesi che riteniamo giusta ed agendo di conseguenza

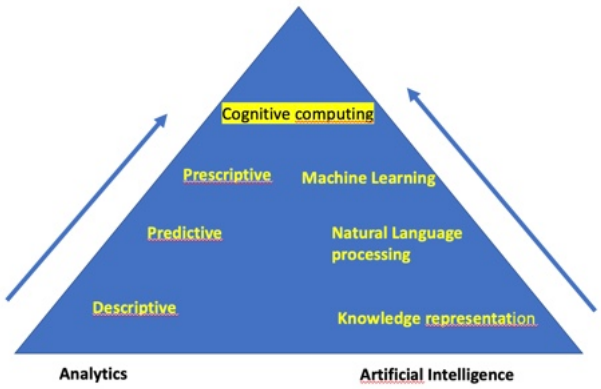
Un sistema cognitivo si comporta allo stesso modo e costituirà, una rivoluzione tecnologica e culturale che sconvolgerà diversi settori in quanto sarà in grado di operare con i big data, adoperando il cloud, IOT e le reti.
I settori di sperimentazione di un sistema cognitivo sono healthcare, travel transportation, economics, retail . Ma un sistema cognitivo può essere usato anche nella didattica ad esempio per personalizzare l'apprendimento e modificare il percorso formativo degli studenti.

Design principles of cognitive computing and design thinking
Il disegno di un Sistema cognitivo richiede molte fasi. Richiede la comprensione dei dati, l'identificazione del tipo di domande che bisogna porre e la creazione di un corpus sufficientemente completo da supportare la generazione di ipotesi, relativamente al dominio sotto esame, di fatti osservati. Insomma un sistema cognitivo è molto diverso da un sistema software perché è disegnato per creare ipotesi a partire da dati, analizzare alternative alle ipotesi stesse e determinare la disponibilità di prove a supporto per risolvere un problema. Bisogna costruire un nuovo modello di processo per le applicazioni cognitive perché debba accedere gestire ed analizzare dati in un determinato contesto applicativo, generi e misuri molte ipotesi (soluzioni) rilascino per ognuno prove a supporto ed indizi sul livello di confidenza, il sistema stesso si aggiorni continuamente non appena si aggiornano i dati in modo da diventare sempre più intelligente. Per progettare un sistema così complesso e con una spiccata personalizzazione sul dominio applicativo è necessario uno strumento di disegno più agile e semplice: il design thinking. In breve il design thinking (Il pensiero progettuale), in una delle sue quattro declinazioni, è un approccio di problem solving finalizzato a migliorare le esperienze delle persone per comprendendere i bisogni dell'utente immaginando più soluzioni possibili per rispondere alle sue esigenze. La metodologia è enormemente creativa e si incontra naturalmente con il cognitive computing. il Design Thinking sta spingendo sempre più imprese a cambiare il loro modo di innovare progettando le applicazioni cognitive in maniera creativa e dualmente si comincia a porsi la domanda: Come possono impattare gli algoritimi di intelligenza artificiale sulle attività di Design Thinking?. UN mini corso di design thinking verrà erogato in questo corso allo scopo di riuscire a progettare una applicazione cognitiva in gruppo ed apprezzare questa metodologia innovativa.

Cognitive computing ecosystem scenario
Si tratteggia l'ecosistema vasto del cognitive computing. Ogni disciplina dirompente ne ha uno

Applying advanced analytics to cognitive computing
Il capitolo affronta un interessante convergenza fra "vecchio e nuovo". Uno studio condotto da IBM Institute for Business Value titolato "Facing the storm: Navigating the global skill crisis, condotto con oxford economics ha interrogato 5600 ceo

rappresentanti di 18 industrie, 48 paesi e 800 leaders di istituzioni governative oltrechè 1500 ricercatori in università e centri di ricerca. Il 74% degli intervistati crede che i modelli di business delle aziende non sono più sostenibili in un mercato attuale ! Vi consiglio questo documento, la ragione? Le aziende si troveranno sempre a rincorrere il mercato anzicchè anticiparlo osservando le esigenze dei clienti. Il modello di maturità dei loro processi è fragile.
Come illustrato nella figura successiva, le aziende stanno sperimentando una progressione nel modello di maturità dell'analisi, che va dall'analisi descrittiva dall'analisi predittiva al machine learning e al cognitive computing. Le aziende che hanno avuto successo sono quelle che hanno capito come analizzare i dati per capire sia dove sono state collocate nel passato ma anche come possono imparare dal passato per anticipare il futuro. Possono descrivere in che modo azioni ed eventi avranno un impatto sui risultati. Sebbene la conoscenza di questa analisi possa essere utilizzata per fare previsioni, in genere queste previsioni sono fatte attraverso una lente di aspettative preconcette. I data scientist e gli analisti aziendali sono stati costretti a fare previsioni basate su modelli analitici basati su dati storici. Tuttavia, ci sono sempre fattori sconosciuti che possono avere un impatto significativo sui risultati futuri. Le aziende hanno bisogno di un modo per costruire un modello predittivo in grado di reagire e cambiare in caso di cambiamenti nell'ambiente. La prossima frontiera, foriera di opportunità e grossi cambiamenti, include i big data analytics ed include le tecnologie del machine learning e del cognitive computing. Come mostrato nella prossima figura c'è una convergenza di tecnologie fra l'analisi (anche convenzionale) e l'intelligenza artificiale. Una spinta importante per questa convergenza è il cambiamento nei tempi e nell'immediatezza dei dati. Le applicazioni odierne richiedono spesso cambiamenti di pianificazione e operativi a un ritmo rapido affinché le aziende rimangano competitive. Attendere 24 ore o più per i risultati di un modello predittivo non è più accettabile. Ad esempio, un'applicazione di gestione delle relazioni con i clienti può richiedere un processo di analisi iterativo che incorpora le informazioni correnti dalle interazioni con i clienti e fornisce risultati a supporto del processo decisionale in pochi secondi, garantendo che il cliente sia soddisfatto. Pertanto, i modelli analitici devono incorporare set di grandi dimensioni che includano dati strutturati, non strutturati e in streaming per migliorare le capacità predittive. La moltitudine di fonti di dati che le aziende devono valutare per migliorare l'accuratezza del modello include database operativi, social media, sistemi di relazione con i clienti, web log, sensori e video.



Unstructured Information Management architecture & Massively parallel probabilistic evidence-based architecture
In questo capitolo si tratta dell'architettura che è stata utilizzata dal team DeepQA per costruire il sistema che ha battuto il migliore giocatore umano nel torneo Jeopardy! . Quindi è, secondo me, l'invenzione più brillante della letteratura sulla QA negli ultimi decenni.
Quindi, il motivo per cui Watson IBM funziona e perché similmente funzionano gli altri sistemi simili è perché l'Architettura DeepQA è stata elaborata in un modo molto flessibile e consente l'integrazione di una varietà di tecnologie diverse tra cui machine learning, elaborazione del linguaggio naturale e reasoning e rappresentazione della conoscenza, e quasi tutto ciò che è possibile pensare nel campo dell'intelligenza artificiale.
E prima di tutto, forse l'innovazione più importante nell'architettura DeepQA è il fatto che possiamo usare simultaneamente sia i dati strutturati che quelli non strutturati nello stessa architettura. L'architettura è ritagliabile, anche questa è una importante innovazione, a seconda del dominio ad esempio possiamo avere una versione semplificata di questa architettura DeepQA che è quella che chiamiamo la pipeline DeepQA minima. L'architettura stessa poi è tipicamente parallela perché essa deve esplorare tutte le strade (in parallelo) in quanto essa ha un approccio probabilistico basato sulle evidenze.

Cooperative computing programming & Flow-based programming
La cooperazione e la collaborazione sono la sfida fondamentale per un gruppo di lavoro anche molto nutrito e decentrato

a livello geografico. Interessanti gli strumenti di programmazione per attivare la cooperazione, distribuire i task, controllare i lavori e coordinare le attività, affinche lo sforzo converga nella direzione di amplificare gli sforzi in una collaborazione del team di sviluppo. Sebbene il progettista di sistemi cognitivi, scrive poco codice deve riusarne parti e coordinare molte librerie questo rende essenziale usare piattaforme mature per lo sviluppo. Eclipse rappresenta una di queste per la sua capacità di ospitare tutti i linguaggi AI (phyton, R, etc) oltre che tutti i linguaggi di programmazione più tradizionali con i quali moti algoritmi sono scritti (c, c++, c#, java, javascript, ruby, etc). Fino ad arrivare ai linguaggi tipicamente adoperati in Ai come python che sarà adoperato in questo corso [4]. Si ha il vantaggio di poter intervenire con un solo IDE per qualunque linguaggio e conservare uno standard di cooperazione durante lo sviluppo. Flow-based programming è il modo più facile ed agevole per unire i vantaggi di un workflow ed un dataflow quando si progetta una applicazione AI. In questo corso si userà Node-RED che è appunto un flow-based programming tool[3].

The role of cloud and distribute computing in CC: PaaS platform as IBM-Watson, Microsoft-Azure, et al
La capacità di sfruttare i servizi di calcolo fortemente distribuiti e a buon mercato non solo ha trasformato il modo con il quale oggi viene gestito e distribuito il software ma è anche diventato un cardine per l'attività di commercializzazione del cognitive computing. Enormi sistemi di elaborazione cognitiva richiedono un ambiente di elaborazione convergente che supporti una varietà di tipi di hardware, servizi software ed elementi di rete che devono essere bilanciati dal carico di lavoro. Pertanto il cloud computing e un'architettura distribuita sono i modelli di base necessari per rendere operativo i sistemi cognitivi su larga scala.
Si passano in rassegna le maggiori Platform as a Service come IBM Watson, Microsoft Azure, Google Ai Platform, etc. Ben sapendo che lo scenario, al momento, mostra almeno una ventina di piattaforme di cognitive computing. Il capitolo prova a delineare criteri di scelta "orientata" ai problemi ed al dominio. In maniera da rendere oggettiva la scelta ma anche osservando altri criteri come i costi, il trattamento dei dati e la loro tenuta e tanti altri criteri che potrebbero risultare prioritari in alcuni contesti.

Building a Watson-enabled system
Si definisce, definite un dominio, un modo originale per ritagliare una architettura di un sistema cognitiva su di un problema, con tutto quello che ciò comporta in termini di bilanciamento delle risorse cloud, di calcolo, di memoria e di scelta della macchina adeguata su cui tutto ciò deve essere implementato e deve essere eseguito. Lo si fa prendendo a riferimento una dei sisstemi cogitivi più diffusi IBM-Watson.

The process of Building a Cognitive application
Costruire una applicazione cognitive non è la stessa cosa che costruire una applicazione software. I sistemi cognitivi sono dirompenti e sconvolgono i paradigmi di progettazione, da ciò nasce l'esigenza di disegnare un modello di processo per lo sviluppo dell'applicazione congnitiva che si diversifica da quello dello sviluppo software in molte parti ed, addirittura, in talune è anche più creativo e completo. La definizione degli obiettivi, la definizione del dominio e la comprensione delle reali intenzioni dell'utente e degli indizi nascosti sono una disciplina molto approfondita nella ingegneria del software, ma la disciplina dell'ingegneria dei sistemi cognitivi necessita di strumenti più semplici e condivisi per le fasi alte di progettazione per questo adopereremo strumenti di design thinking i quali aiutano a condividere le specifiche ed accettare i vincoli di dominio. La definizione delle domande e l'esplorazione degli indizi insieme all'acquisizione delle sorgenti di dati che costituiranno il corpora sarà l'elemento base su cui ruoterà tutto il sistema cognitivo. Il corpora, e questa è una differenza importante, è sempre vivo: una volta creato deve essere continuamente aggiornato perché è su questo che l'applicazione si aggiorna. Infine le tecniche di training e di testing del sistema sono a completare questo capitolo.

Application of cognitive computing to internet of things
Le applicazioni cognitive incontrano l'internet delle cose e la gestione dei dati eterogenei in real time costruendo applicazioni specifiche per la risoluzione di problemi sempre più complessi. La gestione di una città intelligente, la gestione di un applicazione cognitiva per migliorare lo stato di saluto di un paziente ed il suo benessere sono due esempi di applicazioni che passano attraverso la raccolta di dati che provengono da sensori e che devono essere elaborati in tempo utile per consentire ai "decisori" di attivare le giuste decisioni per controllare i problemi.

The operation of AI platform
Per la maggior parte delle organizzazioni, l'integrazione di soluzioni di intelligenza artificiale all'interno dei flussi di lavoro aziendali, rappresenta, ad oggi, un lavoro complesso, costellato di fallimenti . Solo la metà dei progetti di intelligenza artificiale passa dalla fase pilota a quella della produzione. E, i quest'ultimo caso,mediamente servono circa otto-nove mesi per arrivare alla messa a punto completa di un sistema AI efficiente e integrato all'interno della propria operatività, conferendo a tale sistema un valore tangibile nella trasformazione del business. Ebbene, Gartner prevede che, entro il 2025 – complice la sempre maggiore maturità delle tecnologie di AI Orchestration and Automation Platform – il 70% delle

aziende avrà reso operative le proprie architetture di intelligenza artificiale, spostando così i progetti AI dall'idea alla produzione e rendendoli, nel concreto, utili.

Efficient Use of resources
Investire nell'intelligenza artificiale significa utilizzare in modo efficiente tutte le risorse a disposizione, compresi i dati e i modelli di calcolo. Un esempio viene proprio dalla composite AI che, coniuga tecniche diverse, tra cui deep learning, analisi dei grafi, modellazione basata su agenti e tecniche di ottimizzazione. Col risultato di un sistema di intelligenza artificiale "composito", in grado di risolvere una gamma più ampia di problemi aziendali. Ma è necessario che tali tecnologie (e i dati utilizzati per allenare gli algoritmi) vengano usate con la massima efficienza da chi ne possiede le competenze.

Responsible AI
"Una maggiore trasparenza e verificabilità delle tecnologie di intelligenza artificiale continua a rivestire un'importanza cruciale. Questa è l'AI responsabile". Tanto più necessaria quanto più l'AI arriva a sostituire le decisioni umane su larga scala, amplificando gli impatti positivi e negativi di tali decisioni. L'attenzione deve focalizzarsi sui dati che vengono somministrati alla macchina. Se il dato è "buono", scevro da pregiudizi, allenerà un algoritmo altrettanto "buono" e libero da bias. L'attenzione, in particolare, deve essere alta nei confronti dei pregiudizi "impliciti", meno manifesti e, dunque, più difficili da individuare. Come, ad esempio, quelli che conducono a decisioni discriminatorie nei confronti dell'età o del genere. Nei prossimi anni, dunque, le organizzazioni dovranno poter sviluppare e gestire sistemi di intelligenza artificiale "che siamo etici e trasparenti" e, per il conseguimento di tale obiettivo, tutto il personale dedito allo sviluppo di sistemi AI dimostri di possedere esperienza nella "responsible AI".

Approach that combines small data & wide data
In tema di nuovi trend dell'intelligenza artificiale, un'altra tendenza riguarda i "dati piccoli" (small data), vale a dire quei dati che hanno a che vedere con l'applicazione di tecniche analitiche che necessitano di un numero inferiore di informazioni. Tali dati insieme all'utilizzo di grandi set di dati (wide data o big dta), consentono analisi più approfondite e aiutano a ottenere una visione più a ampia del problema che si intende risolvere per mezzo dell'AI. L'Hype Cycle for Artificial Intelligence 2021 indica che, entro il 2025, il 70% delle organizzazioni sarà costretto a spostare la propria attenzione dai dati grandi a quelli piccoli, conferendo, in questo modo, più spazio all'analisi e all'incrocio dei dati stessi. Si tratta, di una tendenza osservata a partire dalla crisi pandemica, che ha causato un rapido decadere delle grandi mole di dati storici, correlati a situazioni passate, e rompendo, così, schemi precedenti. Adottare tecniche di analisi che coniugano "small data" e "wide data" significa, invece, lavorare con volumi diversi di dati ed estrarre valore da fonti diverse e non strutturate.

MATERIALE DIDATTICO

Libri di testo:

- E. Kelly III and S. Hamm, Smart Machines IBM's Watson and the Era of Cognitive Computing (2014), ISBN: 978-0-231-16856-4.
- Alfio Gliozzo et al, Building Cognitive Applications with IBM Watson Services: Volume 1 Getting Started, (2017) IBM redbook , https://ibm.co/30V63PU
- J. Hurwitz, M. Kaufman, A. Bowles, Cognitive Computing and Big Data Analytics (2015), ISBN: 978-1-118-89662-4.
- Taiji Hagino, Practical Node-RED Programming (March 2021), ISBN 978-1-80020-159-0.
- Paul Deitel, Harvey Deitel, Intro to Python for Computer Science and data Science, (2020), ISBM-13: 978-0-13-540467-6.

Paolo Maresca Materiale delle lezioni e registrazioni del corso a.a. 2020-2021 COGNITIVE COMPUTING SYSTEMS accesso su Teams con codice tr9xxpl link: https://bit.ly/2SBD0t3

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà lezioni frontali per circa la metà (50%) del corso, la restante parte sarà dedicato (50%) alle esercitazione, seminari ed ai laboratori. I laboratori serviranno ad approfondire gli aspetti teorici e metodologici emersi nelle lezioni frontali. Data la spiccata correlazione del corso con l'evoluzione del cognitive computing, i laboratori potranno essere diversi di anno in anno e saranno chiamati anche esperti di aziende a tenere le tematiche più innovative. Seminari sulle nuove tendenze del cognitive computing, sono altresì previsti.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

SCHEDA DELL'INSEGNAMENTO (SI)

"BUSINESS PROCESS AUTOMATION"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: VALERIA VITTORINI
TELEFONO: 0817683847
EMAIL: VALERIA.VITTORINI@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): II
SEMESTRE (I, II): II
CFU: 3

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Per una migliore fruizione dei contenuti del corso lo studente dovrebbe avere conoscenze base di programmazione e dei linguaggi XML e Java.

OBIETTIVI FORMATIVI

Obiettivo del corso è fornire agli studenti i principali concetti relativi al workflow management. Il focus del corso è sulla definizione, rappresentazione e codifica del workflow, mediante l'utilizzo di linguaggi, quali Business Process Modeling Notation (BPMN), e workflow patterns. Casi di studio vengono presentati relativi a moderni sistemi basati su cloud e servizi (e.g., Netflix, AWS).

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di aver acquisito le conoscenze di base necessarie alla definizione e alla gestione di workflow in diversi domini applicativi.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare autonomia nell'utilizzo di tecnologie e strumenti nella definizione ed esecuzione di semplici processi di business, attraverso uno o più piattaforme introdotte durante il corso.

PROGRAMMA-SYLLABUS

BPM e Workflow. Inquadramento nel contesto del Business Process Management (BPM). Definizione, classificazione dei processi di business, ciclo di vita. Workflow: BPM e workflow, definizione di workflow, concetti fondamentali, architettura di riferimento di un workflow management system. Process model: eventi, trigger, operatori di routing. Workflow data. Modello organizzativo, classificazione delle risorse. Linguaggi per la definizione di processi di business.

Orchestrazione e Coreografia. SOA, REST, microservices: principi fondamentali. Automazione mediante orchestrazione di servizi.

Automazione e Application Integration: Livelli. Pattern e modelli di integrazione. Componenti fondamentali: Adapters, wrappers, brokers. Architetture per l'integrazione di applicazioni: Hub and Spoke, Bus, Middleware, Enterprise Service Bus. Workflow Patterns.

Casi di studio. Netflix, AWS.

MATERIALE DIDATTICO

Appunti del corso, articoli scientifici, documentazione applicazioni utilizzate come caso di studio.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Lezioni frontali (40%), esercitazioni (30%), attività seminariali (30%)

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	
discussione di elaborato progettuale	x
altro	

SCHEDA DELL'INSEGNAMENTO (SI)

"IMAGE PROCESSING FOR COMPUTER VISION"

SSD ING-INF/03

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: GIUSEPPE SCARPA
TELEFONO: 0817683768
EMAIL: GIUSEPPE.SCARPA@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO: I/II
SEMESTRE: I
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Teoria dei Segnali

OBIETTIVI FORMATIVI

L'insegnamento si propone di fornire agli studenti nozioni approfondite sullo sviluppo e l'applicazione di tecniche di elaborazione delle immagini per la soluzione di tipici problemi di *computer vision*, spaziando da metodi tradizionali per l'elaborazione dei segnali, cioè orientati alla modellizzazione, ad approcci moderni basati su reti neurali convoluzionali. Specifici problemi di *computer vision* considerati quali obiettivi formativi del corso sono la rivelazione, caratterizzazione ed il *matching* di *feature* locali, il *fitting* e l'allineamento di modelli geometrici, la classificazione di immagini, la segmentazione semantica o per istanze di immagini, la rivelazione, localizzazione ed il riconoscimento degli oggetti, la stima della posa, la stima della profondità, la corrispondenza stereo, la ricostruzione 3D da viste multiple.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente dovrà conoscere sia tecniche di filtraggio classiche che approcci basati su moderne reti neurali convoluzionali per la soluzione di problemi di visione computazionale quali la rivelazione, descrizione ed il *matching* di *feature* locali, il *fitting* e l'allineamento di modelli geometrici, la classificazione di immagini, la segmentazione semantica o per istanze, la rivelazione, localizzazione e riconoscimento di oggetti, la stima della posa, la stima della profondità, la corrispondenza stereo, la ricostruzione 3D da viste multiple, dalla prospettiva dell'elaborazione dei segnali. Per i problemi elencati lo studente dovrà altresì conoscere le metriche o gli indici prestazionali utili alla valutazione delle possibili soluzioni.

Capacità di applicare conoscenza e comprensione

Lo studente deve acquisire la capacità di progettare, sviluppare e testare algoritmi di elaborazione delle immagini allo stato dell'arte finalizzati alla risoluzione di comuni problemi di visione computazionale, tra cui la rivelazione, descrizione ed il *matching* di *feature* locali, il *fitting* e l'allineamento di modelli geometrici, la classificazione di immagini, la segmentazione semantica o per istanze, la rivelazione, localizzazione e riconoscimento di oggetti, la stima della posa, la stima della profondità, la corrispondenza stereo, la ricostruzione 3D da viste multiple.

PROGRAMMA-SYLLABUS

Richiami sul filtraggio delle immagini. Dominio spazio-scala e decomposizione piramidale. Richiami sugli ambienti di programmazione per lo sviluppo di algoritmi di *computer vision*.

Formazione dell'immagine: La luce e il colore. Il modello *pinhole camera*. La proiezione del mondo 3D nel piano dell'immagine: matrice di proiezione della camera e calibrazione della camera. Trasformazioni geometriche di tipo proiettivo.

***Early vision*:** Rivelazione dei contorni; segmentazione mediante trasformata *watershed*; *template matching* e descrizione tessiturale; rivelazione di angoli (Harris *detector*) e linee (trasformata di Hough).

Rivelazione e descrizione di *keypoint*: Definizione di *keypoint* e proprietà di ripetitività. Proprietà di invarianza dei rivelatori rispetto ad illuminazione, traslazione, rotazione, scala, trasformazioni affini e omografie. Rivelatore di Harris. Differenza di gaussiane (DoG). Piramide di DoG. Orientazione e scala di un *keypoint*. Descrittori di *feature*: proprietà discriminative; descrittori di comune impiego (SIFT, SURF, MSER,...); descrittori di forma e contesto.

***Matching*, *fitting* ed allineamento:** *Matching* di *feature* mediante criterio del rapporto delle distanze. *Fitting* ed allineamento: metodo dei minimi quadrati lineare o robusto; algoritmo ICP; trasformata di Hough generalizzata; algoritmo RANSAC. Rivelazione, riconoscimento e classificazione.

Elaborazione delle immagini mediante reti neurali convoluzionali (CNN): Architetture convoluzionali per l'elaborazione delle immagini. Addestramento di CNN per l'elaborazione delle immagini: backpropagation e algoritmo di ottimizzazione SGD (e varianti). Moduli (*layer*) di comune impiego: convoluzione, *pooling, unpooling*, *batch normalization*, funzioni di attivazione (ReLU e sue varianti, Tanh, sigmoide). Funzioni di costo per l'elaborazione delle immagini. *Dropout* e data *augmentation*. Modelli CNN per super-risoluzione, classificazione, segmentazione, rivelazione e localizzazione di oggetti, stima della profondità, stima della posa.

Visione *multi-view*: Visione stereo: disparità e profondità. Vincoli epipolari; matrice essenziale e matrice fondamentale. Problemi di corrispondenza densi. Ricostruzione 3D da *multi-view*: *Structure from Motion* (SfM).

MATERIALE DIDATTICO

- R. Szeliski, "Computer vision: algorithms and applications", Springer 2010.
- R.-I. Hartley, A. Zisserman, "Multiple View Geometry in Computer Vision", C. U. P., 2nd Ed., 2004.
- I. Goodfellow, et al., "Deep Learning", MIT Press, 2017.
- Dispense del docente.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso prevede sia lezioni frontali (circa il 60% del totale) che attività di laboratorio. Sono tra l'altro previsti tutorial introduttivi sia sul linguaggio di programmazione Python e gli annessi *toolbox* per il *deep learning* che sull'uso di piattaforme di calcolo in *cloud* funzionali agli obiettivi del corso. Parte delle ore di laboratorio saranno dedicate allo sviluppo in itinere, con tutoraggio, dei progetti degli studenti ai fini della valutazione finale.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	
discussione di elaborato progettuale	
altro	x

L'esame prevede la presentazione di un progetto svolto individualmente o in gruppo, con relativa discussione, ed un colloquio generale sui contenuti del corso. Il progetto è di norma sviluppato in itinere e presentato al termine del corso in un *workshop* di chiusura, mentre il colloquio può tenersi in qualsiasi appello dell'a.a. corrente senza vincoli temporali.

SCHEDA DELL'INSEGNAMENTO (SI)

"ECONOMIA E ORGANIZZAZIONE AZIENDALE"

SSD ING-IND/35*

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ANNA RUGGIERO
TELEFONO:
EMAIL: ANNA.RUGGIERO2@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II): I/II
SEMESTRE (I, II): II
CFU:9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Conoscenze basilari di Analisi Matematica I

OBIETTIVI FORMATIVI

Obiettivo del corso è fornire agli studenti conoscenze relative ai principi, alle best practices, alle regolamentazioni e ai L'insegnamento si propone di fornire i concetti e gli strumenti fondamentali per comprendere il funzionamento di un sistema economico ed organizzativo partendo dalla prospettiva microeconomica, e fornendo successivamente la prospettiva strategica dell'organizzazione sia dal punto di vista esterno che interno.
Per quanto concerne la prospettiva microeconomica, saranno oggetti di analisi ed approfondimento i principali problemi relativi al comportamento dei consumatori e delle imprese, con un focus sulle principali forme di mercato.
Per quanto concerne la prospettiva dell'organizzazione, verrà approfondita la condizione esterna attraverso l'apprendimento degli strumenti di analisi dell'ambiente esterno per la definizione della strategia aziendale, e verranno introdotti i principi del business model e del business plan.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente dovrà acquisire il lessico necessario a descrivere i fenomeni economici. Egli dovrà dimostrare di conoscere i concetti fondamentali della teoria del consumatore e del produttore. Lo studente dovrà dare prova di comprendere come le organizzazioni definiscono la strategia aziendale sulla base delle analisi effettuate sull'ambiente in cui l'organizzazione vive.

Capacità di applicare conoscenza e comprensione

Sulla base delle conoscenze acquisite, lo studente dovrà essere in grado di analizzare il funzionamento di una organizzazione attraverso la simulazione di un progetto di startup in cui verranno applicati tutti i concetti appresi.

PROGRAMMA-SYLLABUS

Problemi e tecniche associati all'analisi, alla progettazione, allo sviluppo e alla verifica di sistemi utilizzati in INTRODUZIONE ALL'IMPRESA (Definizione di impresa, costi e obiettivi di impresa, i criteri di classificazione delle imprese)
MICROECONOMIA (L'impresa ed il mercato, la domanda e l'offerta, le forme di mercato)
STRATEGIA (Definizione di settore; analisi di settore –modello di PORTER; ciclo di vita del settore; Strategie concorrenziali di base; Posizionamento dell'impresa –analisi SWOT)
BUSINESS PLAN (executive summary, piano operativo, piano organizzativo, piano marketing)

MATERIALE DIDATTICO

- Dispense e materiali messe a disposizione dal docente
- "Il mestiere di fare impresa. Imprenditori non si nasce, si diventa!". Di Michele Raffa, Edizioni Scientifiche Italiane

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Lezioni frontali, esercitazioni, seminari esterni, simulazione di un progetto di impresa (lavoro di gruppo)

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

b) Modalità di valutazione

Valutazione individuale in trentesimi sulla prova scritta (obbligatoria)
Valutazione di gruppo sulla presentazione del progetto

SCHEDA DELL'INSEGNAMENTO (SI)

"SAFETY-CRITICAL SYSTEMS"

SSD ING-INF/05*

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: VALERIA VITTORINI
TELEFONO: 0817683847
EMAIL: VALERIA.VITTORINI@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III):I/II
SEMESTRE (I, II):
CFU:3

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Obiettivo del corso è fornire agli studenti conoscenze relative ai principi, alle best practices, alle regolamentazioni e ai processi per la progettazione dei sistemi safety-critical, con particolare riferimento alla modellazione e all'analisi di tali sistemi e alle tecniche di verifica formale.
Verranno illustrati il ruolo e l'importanza dei metodi formali nello sviluppo di sistemi safety-critical e verranno introdotti diversi strumenti formali utilizzati per la modellazione di sistemi e di proprietà. Infine verranno affrontati aspetti avanzati in particolare nell'ambito delle metodologie di modellazione di sistemi complessi.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e comprendere i concetti alla base dello sviluppo di sistemi safety-critical e il ruolo dei modelli formali per l'analisi di sistemi critici, e in particolare per la verifica di proprietà e per la validazione.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di saper applicare i concetti e le tecniche presentati nel corso a semplici casi di studio, anche sviluppando e analizzando semplici modelli formali.

PROGRAMMA-SYLLABUS

Parte I: Problemi e tecniche associati all'analisi, alla progettazione, allo sviluppo e alla verifica di sistemi utilizzati in applicazioni critiche per la sicurezza. Elementi di progettazione e sviluppo di sistemi safety-critical (architetture, tecniche per lo sviluppo si sistemi safety-critical, e.g.; formal testing, high-integrity programming, ispezione del software), standard europei.
Parte II: Il ruolo dei metodi formali nell'ingegneria dei sistemi, i metodi formali nella certificazione dei sistemi reali, alcuni esempi tratti dal mondo reale, proprietà funzionali e non funzionali, analisi qualitativa e quantitativa. Petri nets ed estensioni tempificate per l'analisi quantitativa di proprietà temporali, linguaggi formali per la specifica e l'analisi. Logiche temporali, LTL e CTL, introduzione al model checking. tecniche di sviluppo di modelli complessi, strumenti per la modellazione e la risoluzione dei modelli.
Parte III: esercitazioni e applicazione a casi di studio.

MATERIALE DIDATTICO

Appunti del corso, articoli scientifici.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Lezioni frontali (60%), esercizi e attività di laboratorio (30%), seminari applicativi (10%).

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità d'esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	
discussione di elaborato progettuale	x
Altro	

SCHEDA DELL'INSEGNAMENTO (SI)

"CIRCUITI PER DSP"

SSD ING-INF/01

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: DAVIDE DE CARO
TELEFONO: 081-7683136
EMAIL: DAVIDE.DECARO@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): I/II
SEMESTRE (I, II): I
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuna

EVENTUALI PREREQUISITI

Conoscenza di base del funzionamento dei circuiti digitali e del linguaggio C.

OBIETTIVI FORMATIVI

L'insegnamento si pone l'obiettivo di fornire agli studenti le nozioni avanzate e di alto livello relative alla implementazione degli algoritmi di elaborazione digitale dei segnali e, inoltre, di acquisire le conoscenze, sia di base sia avanzate, relative alle architetture dei circuiti DSP disponibili commercialmente, delle problematiche, sia teoriche che pratiche, relative alla implementazione ottimale, in tempo reale, su DSP, dei principali algoritmi di elaborazione digitale dei segnali e dell'ambiente di sviluppo per la loro programmazione.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Al termine del processo di apprendimento lo studente è edotto sulle principali tecniche di elaborazione dei dati in virgola fissa e mobile, sugli strumenti di analisi degli errori di rappresentazione e delle problematiche di overflow in un sistema lineare realizzato in aritmetica a virgola fissa, sulle principali tecniche di prevenzione/gestione dell'overflow, sulle problematiche teoriche relative alla descrizione in linguaggio C di algoritmi di elaborazione digitale dei segnali, sulle caratteristiche architetturali di minima che identificano un circuito DSP (Digital Signal Processor), sulle problematiche relative alla pipeline nei DSP, sia nel caso di pipeline protetta che nel caso di pipeline non-protetta, sulle caratteristiche dei DSP ad elevato parallelismo di elaborazione -in particolar modo basati su architettura di tipo Very Long Instruction Word (VLIW)-, alle forme di parallelismo al livello dell'instruction set del DSP -istruzioni Single-Instruction-Multiple-Data-, sulle tecniche di ottimizzazione del codice per le architettura VLIW -in particolar modo utilizzando tecniche di Loop Unrolling e Software Pipelining-, sulle caratteristiche di base e avanzate dei sistemi di interruzione e DMA (Direct-Memory-Access) nei DSP, sulle caratteristiche delle interfacce di comunicazione del DSP -in particolar modo quelle di tipo seriale sincrono-, sulle problematiche relative al realizzazione su un DSP di sistemi di elaborazione in tempo reale -in particolar modo utilizzando l'elaborazione in streaming oppure a blocchi-.

Capacità di applicare conoscenza e comprensione

Al termine del processo di apprendimento lo studente è in grado di progettare, autonomamente, l'implementazione in aritmetica a virgola fissa e mobile, di un algoritmo di elaborazione dei segnali tenendo conto sia degli errori di rappresentazione che delle problematiche connesse all'overflow e di determinare l'approccio di prevenzione/gestione dell'overflow che meglio si adatta alla specifica applicazione.
Lo studente acquisisce inoltre la capacità di descrivere in linguaggio C un algoritmo di elaborazione digitale dei segnali con particolare riferimento alle problematiche relative alla efficiente implementazione su DSP sia a virgola fissa che a virgola mobile. Lo studente padroneggia inoltre le tecniche di ottimizzazione del codice, con particolare riferimento alle architetture VLIW ed è in grado di gestire nel modo migliore possibile, in relazione alla specifica applicazione, il trade-off tra tempo di calcolo e dimensione del codice applicando le tecniche di Loop Unrolling e Software Pipelining.
In fine, lo studente padroneggia gli approcci per l'elaborazione in tempo reale di un algoritmo di elaborazione digitale su un DSP, e, in relazione alle caratteristiche della specifica applicazione, può risolvere nel modo migliore possibile il trade-off tra massima frequenza di elaborazione e latenza adottando approcci di elaborazione in streaming oppure a blocchi. Lo studente inoltre padroneggia le problematiche inerenti l'impiego delle unità DMA sia a supporto dell'elaborazione, sia a servizio delle interfacce del DSP nella elaborazione a blocchi.

Autonomia di giudizio, Abilità comunicative, Capacità di apprendimento

In relazione a tutte le capacità descritte in precedenza, lo studente acquisisce inoltre capacità autonoma di giudizio essendo, questa, tra l'altro, nel particolare contesto, un presupposto imprescindibile dell'attività creativa di tipo progettuale che rientra nelle capacità di applicare la conoscenza che sono sviluppate.

Durante l'insegnamento gli studenti vengono inoltre stimolati nella acquisizione degli strumenti che consentono l'approfondimento in modo autonomo degli argomenti trattati, mentre le metodologie di verifica dell'apprendimento da parte degli studenti tendono a sviluppare anche le loro abilità comunicativa.

PROGRAMMA-SYLLABUS

Introduzione: Circuiti per l'Elaborazione Digitale: ASIC, FPGA, DSP. Elaborazioni Real-Time.

Problematiche numeriche nella rappresentazione dei segnali: Richiami sulle rappresentazione Fixed-Point e Floating-Point; Precisione ed Errori di Quantizzazione; Errori di Quantizzazione al livello di Sistema; Overflow e tecniche di prevenzione: Variazione della Rappresentazione, Scaling, Scaling per segnali a banda stratta, Scaling Statistico, Aritmetica Saturata, Bit di Guardia; Applicazione delle tecniche di prevenzione dell'Overflow ad un Filtro FIR. Descrizione degli algoritmi di DSP in linguaggio C: rappresentazione Fixed-Point Generalizzata, Regole di Promozione del Tipo, Esempio (filtro FIR con tecnica dei bit di Gaurdia).

Architetture di base dei circuiti DSP: Datapath; Architettura Harvard ed Harvard modificata; Memoria nei DSP: Banchi Separati, Banchi in Interveaving, Memorie Dual-ported, Allineamento dei Dati ed Accessi Multipli, Gerarchie delle memorie, Caching coerenza e predicibilità; Repeat Buffer; Indirizzamenti per DSP ed Unità di generazione degli indirizzi: indirizzamento Circolare e Bit-Reversal; Tecniche di Zero-Overhead-Looping; Semplici periferiche nei DSP: Timers/PWM generators; Esempi di DSP della Texas Instruments: architetture C2xx, C54xx e C55xx.

Architetture avanzate dei DSP: Trade-off tra Ortoganalità e Dimensione del Codice; Pipelining nei DSP: Pipeline non-Protetta o Visibile, Pipeline Protetta o Trasparente, Hazards Strutturali, Hazard sui Dati e Dipendenze, Hazard di Controllo, Delayed-Branch; Scheduling Statico e Dinamico delle Istruzioni; Architetture Superscalari (cenni); Architetture Very-Long-Instruction-Word (VLIW); Pipeline non-Protetta ed Interruzioni (trade-off tra interrombibilità del codice e tempo di calcolo); Istruzioni ed Aritmetica Single-Instruction-Multiple-Data (SIMD); DSP VLIW della Texas Instruments: architetture C64xx, C67xx, C66xx. Cenni ai principali DSP di Analog Devices (Blackfin, SigmaDSP, SHARC e TigerSHARC) e Freescale (Symphony, StarCore SC3400 e SC3850). DSP con Multi-threading Hardware: architettura Hexagon di Qualcomm. SoC basati su DSP di Texas Instruments: DaVinci Digital Media Processor, OMAP, Keystone e Keystone II.

Architettura dei DSP Texas Instruments C64xx e C67xx: Architettura VelociTI; Set Istruzioni; Architettura delle Memoria e Caching su due livelli; Esecuzione Condizionale; Indirizzamenti con offset; Indirizzamenti Circolari e Registro AMR; Pipeline del DSP: Delay-Slots e Latenza delle istruzioni; Istruzioni SIMD.

Sviluppo del Codice ed Ottimizzazione per DSP VLIW: Sviluppo in Assembly: Grafi delle Dipendenze, Parallelizzazione delle Istruzioni, Eliminazione dei NOP, Loop Unrolling; Software Pipelining: Minimum-safe-trip-count, Resource-Bound, Loop-Carried-Rosource-Bound, Esecuzione Speculativa, Problemi di Live-too-Long, Utilizzo congiunto di Loop Unrolling e Software Pipelining; Sviluppo del codice in Linear-Assembly; Sviluppo del Codice in C: direttive per il Software Pipelining e l'Unrollimg, Aliasing dei Puntatori, Direttive per l'Allocazione in Memoria, Livelli di Ottimizzazione, Funzioni Static, Interrompibilità del Codice, Software Pipelined Loop Buffer (SPLOOP); Linker, Variabili Globali e Static, Allocazione delle variabili Near e Far, Start-up del DSP.

Interrupt e DMA nei DSP Texas Instruments C6x: Interrupt: funzionalità di base, registri per la gestione degli interrupt, funzionamento dettagliato in Hardware, scrittura della Interrupt-Service-Routine (ISR), tempi caratteristici; DMA: funzionalità di base, parametri e tipologie di trasferimento, sincronizzazione, Quick-DMA, generazione di interrupt di completamento, esempi.

Implementazione Real-Time degli Algoritmi di DSP: Interfacce Seriali Sincrone ed interfaccia McBSP dei DSP Texas C6x; Gestione delle comunicazione in Polling, Interrupt, DMA; Elaborazioni in Streaming di tipo Interrompibile e non-Interrompibile, Controllo di Flusso nelle elaborazioni in Ricezione/Trasmissione; Sistemi Operativi Real-Time e Elaborazioni con Prelazione; Elaborazioni a Blocchi: Ping-Pong Buffering; Tecniche di Buffering per gli Algoritmi con Memoria; Debugging off-line e real-time nei DSP, Tecniche di In-System Debugging tramite interfaccia JTAG.

Esercitazioni: Le esercitazioni sfruttano l'ambiente di sviluppo Code Composer Studio di Texas Instruments e le schede per la prototipizzazione rapida su DSP DSK-C6713.
Implementazioni di Filtri FIR: Calcolo dei Coefficienti, Progetto Numerico e Problematiche di Overflow, Ottimizzazione del Codice con esempi di Unrolling, Software Pipelining, Buffering dei dati; Prove sperimentali con I/O in Real-time dal Codec audio presente sulla board DSK-C6713: filtro FIR, equalizzatore audio a 5 bande, equalizzatore audio a 5 bande mediante Sistema Operativo real-time, equalizzatore audio a 9 bande mediante Sistema Operativo real-time.

MATERIALE DIDATTICO

- John G. Proakis, Dimitris G. Manolakis, “Digital Signal Processing: Principles, Algorithms and Applications”, 4° edition, Prentice Hall 2007
- Sen M. Kuo, Woon-Seng Gan, “Digital Signal Processors: Architectures, Implementations, and Applications”, Prentice Hall 2005
- Appunti delle lezioni
- Testi delle esercitazioni

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Lo svolgimento dell’insegnamento prevede lezioni frontali, esercitazioni e, compatibilmente con gli aspetti organizzativi, lezioni di laboratorio.
Per lo svolgimento delle esercitazioni gli studenti adottano l’ambiente di sviluppo Code Composer Studio della Texas Instruments. Il laboratorio, oltre a Code Composer Studio, prevede l’impiego di board per il prototyping rapido (DSK6713 della Texas Instruments) e la strumentazione di base di un laboratorio di elettronica (oscilloscopio, generatore di segnali).

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	✓
discussione di elaborato progettuale	
altro (discussione esercitazioni)	✓

SCHEDA DELL'INSEGNAMENTO (SI)

"MODELLI E ALGORITMI DI OTTIMIZZAZIONE"

SSD MAT/09

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: STERLE CLAUDIO
TELEFONO: 081 768 5911
EMAIL: CLAUDIO.STERLE@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO: I/II
SEMESTRE: II
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Il corso ha l'obiettivo di fornire agli studenti conoscenze avanzate di programmazione matematica per la modellazione e risoluzione esatta di problemi decisionali complessi di ottimizzazione su rete in ambito ingegneristico. Lo studio teorico dei principali algoritmi per il calcolo della soluzione ottima dei problemi decisionali affrontati è completato dalla sperimentazione numerica di tali algoritmi mediante l'utilizzo di software di ottimizzazione. Al termine del corso lo studente avrà acquisito la conoscenza di metodologie avanzate per la modellazione e soluzione di problemi di ottimizzazione continua, intera e mista-intera su reti informatiche e di telecomunicazioni.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il percorso formativo ha l'obiettivo di fornire agli studenti le metodologie di ottimizzazione continua, intera e mista-intera necessarie per la modellazione e risoluzione esatta di problemi ingegneristici nell'ambito delle reti informatiche e di telecomunicazioni. Lo studente deve dimostrare di aver acquisito gli strumenti necessari a formulare un problema decisionale mediante un modello di programmazione matematica, individuando funzione obiettivo, variabili decisionali e vincoli del sistema oggetto di studio. Lo studente deve inoltre essere in grado di individuare il miglior metodo risolutivo da utilizzare per la determinazione della soluzione ottima di un problema decisionale, in relazione alle sue specifiche caratteristiche. Infine, lo studente deve essere in grado di analizzare la sensibilità della soluzione ottenuta rispetto alla variabilità delle condizioni al contorno e comprenderne i nessi causali.

Capacità di applicare conoscenza e comprensione

Il percorso formativo è orientato a trasmettere gli strumenti metodologici e operativi necessari ad applicare concretamente le conoscenze di programmazione matematica a problemi di ottimizzazione su rete emergenti nel contesto informatico e delle telecomunicazioni. In particolare, lo studente deve dimostrare di saper sviluppare tutte le fasi di un processo decisionale: analisi del sistema (definizione delle sue componenti, dei parametri che lo caratterizzano, assunzioni e specifiche di funzionamento); definizione del problema decisionale; selezione/costruzione di un modello matematico di simulazione del sistema; implementazione e risoluzione del modello tramite un algoritmo e/o un software di ottimizzazione; analisi ed interpretazione dei risultati al fine di verificare la qualità della soluzione e mettere in atto eventuali meccanismi di retroazione.

PROGRAMMA-SYLLABUS

- Ottimizzazione non lineare multidimensionale non vincolata
    - Metodi di gradiente
    - Algoritmo di discesa e salita ripida, gradiente coniugato
    - Analisi grafica ed esercitazioni numeriche

- Ottimizzazione non lineare e lineare multidimensionale vincolata
    - Condizioni di ottimo nei problemi di ottimizzazione vincolata (condizioni di Kuhn-Tucker)
    - Rilassamento Lagrangiano
    - Metodi a direzione ammissibile
    - Analisi grafica ed esercitazioni numeriche
    - Ottimizzazione lineare come caso particolare della Ottimizzazione non lineare
    - Algoritmo del simplesso, analisi di stabilità e dualità

- Metodi avanzati di ottimizzazione lineare intera (PLI)
    - Formulazione di problemi ottimizzazione lineare intera e nocciolo convesso
    - Metodi avanzati di risoluzione basati su “row e column generation”
    - Branch and Bound e Branch and Cut
    -Tecniche di rilassamento

- Problemi avanzati di instradamento, localizzazione e progetto su rete (modellazione e soluzione).
    - Problemi di percorso: problemi di minimo percorso e minimo percorso vincolato (minimi percorsi attraverso specificati vertici, con finestre temporali, con risorse limitate), minimo percorso con vincoli di capacità (Quickest path e quickest flow); problema del percorso massimo

    - Problemi di flusso: dei problemi di flusso single e multi-commodity con costi costanti e costi variabili; problemi di massimo flusso; problemi con flussi unicast e multicast;

    - Problemi di network design: problemi di localizzazione; progettazione di reti multi-layer; progettazione e dimensionamento di reti di comunicazione; progettazione di reti affidabili e resilienti.

- Software di ottimizzazione:
    - introduzione ai software di ottimizzazione (Xpress, Cplex);
    - modellazione e risoluzione di problemi reali di programmazione lineare continua, intera e mista-intera tramite tecniche esatte

MATERIALE DIDATTICO

- A. Sforza, Modelli e Metodi della Ricerca Operativa, III ed., ESI, Napoli

- F. S. Hillier, G. J. Lieberman, Ricerca operativa - Fondamenti, 9/ed., McGraw-Hill

- C. Guéret, C. Prins, M. Sevaux, Applications of optimization with Xpress-MP, Editions Eyrolles,Paris

- IBM ILOG CPLEX V12.7 User's Manual for CPLEX

- Materiale didattico integrativo fornito durante il corso e materiale disponibile on-line

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà: lezioni frontali (60%), seminari (10%), esercitazioni di tipo numerico e di introduzione all’uso di software di ottimizzazione (30%). Il materiale del corso sarà reso disponibile on-line agli studenti.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

L’esame prevede lo svolgimento di un elaborato progettuale in cui lo studente deve sviluppare e risolvere un modello di programmazione matematica (continua, intera o mista-intera, rappresentativo di un problema decisionale reale. Il modello sviluppato deve essere implementato in un software di ottimizzazione (Xpress o Cplex) e risolto tramite l’utilizzo delle relative librerie di metodi esatti. L’elaborato progettuale viene assegnato allo studente prima della fine del corso e dovrà essere consegnato prima del colloquio orale. Il colloquio orale avrà come oggetto sia la discussione dell’elaborato progettuale che l’accertamento dell’acquisizione dei concetti e delle metodologie illustrati durante le lezioni.

b) Modalità di valutazione:

La consegna dell’elaborato progettuale è vincolante ai fini dell’accesso al colloquio orale. L’elaborato progettuale e la prova orale contribuiscono ognuna per il 50% della valutazione finale. La consegna dell’elaborato progettuale non è sufficiente per il superamento dell’esame.

SCHEDA DELL'INSEGNAMENTO (SI)

"RISK ASSESSMENT"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ALESSANDRA DE BENEDICTIS
TELEFONO:
EMAIL: ALESSANDRA.DEBENEDICTIS@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): I/II
SEMESTRE (I, II): II
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Il corso ha l'obiettivo di introdurre il processo, le principali metodologie e le tecniche per la valutazione del rischio in sistemi critici.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e comprendere le problematiche relative all'identificazione, valutazione e gestione del rischio in diversi contesti safety e security-critical.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di applicare le principali tecniche di risk assessment viste al corso a casi di studio reali.

PROGRAMMA-SYLLABUS

INTRODUZIONE E TERMINOLOGIA: Definizioni di rischio; hazard scenarios; hazard e threat; hazardous events; fault e failure; probabilità e conseguenze; spettro delle conseguenze; barriere e fattori di escalation. Modello BowTie.

PROCESSO DI GESTIONE DEL RISCHIO: Requisiti e caratteristiche di un processo di gestione del rischio. Panoramica delle principali fasi del processo di gestione del rischio. Gestione del rischio e processo decisionale: processo decisionale deterministico, processo decisionale basato sul rischio; processo decisionale informato sul rischio. La gestione del rischio nella normativa in materia di safety e security.

PROCESSO DI VALUTAZIONE DEL RISCHIO: Panoramica delle principali fasi del processo di valutazione del rischio. Definizione dell'oggetto di studio: il sistema, boundaries di un sistema; system breakdown.

CRITERI DI ACCETTAZIONE DEL RISCHIO: Principi di accettazione del rischio: equità, utilità, tecnologia. L'approccio ALARP: principi ed esempi. Panoramica di altri approcci: ALARA, SFAIRP, GAMAB, MEM. Analisi costi-benefici: valutazione di costi e benefici.

MISURE DI RISCHIO: metriche e misure di rischio individuale e di gruppo: Average Individual Risk (AIR), Location-specific Individual Risk (LSIR), Lost-time injury (LTI) frequency, Lost workdays frequency (LWF); Potential Loss of Life (PLL); Fatal Accident Rate (FAR), FN curves; Risk matrices; Risk Priority Number (RPN).

METODI DI IDENTIFICAZIONE DEGLI HAZARD: Fattori causali di un hazard; Tipi di analisi di safety risk: CD-HAT, PD-HAT, DD-HAT, SD-HAT, OD-HAT, HD-HAT, RD-HAT. Tecniche CD-HAT: metodi basati su checklist e liste di rischio preliminari (PHL). Tecniche PD-HAT: preliminary hazard analysis (PHA); HAZID. Tecniche DD-HAT: subsystem hazard analysis (SSHA); HAZOP. Tecniche SD-HAT: system hazard analysis (SHA); Analisi dei rischi operativi e di supporto (O&SHA); Failure mode, effect and criticality analysis (FMECA): identificazione di modalità, tassi, cause, effetti di guasto. Analisi di criticità qualitativa e quantitativa mediante reti di Petri.

METODI DI ANALISI CAUSALE E DI FREQUENZA: Fult tree analysis: elementi dei diagrammi FT e loro utilizzo nell'ambito del risk assessment. Diagramma di causa ed effetto; Reti bayesiane e loro utilizzo nel risk assessment; Common cause failure analysis.

SVILUPPO DEGLI HAZARD SCENARIOS: Event-trees: pivotal events, passaggi metodologici; cause-consequences analysis.

SECURITY RISK ASSESSMENT: Panoramica dei concetti di rischio per la sicurezza: principali proprietà di sicurezza, minacce, vulnerabilità, debolezze, attacchi, controlli di sicurezza. Rischio per la sicurezza: probabilità e impatto.

MODELLAZIONE DELLE MINACCE: modellazione delle minacce e threat intelligence; MITRE ATT&CK framework. Modellazione di sistemi per il threat modeling: Data Flow Diagrams (DFD) e Process Flow Diagrams (PFD). Metodologie e framework di identificazione e valutazione delle minacce: la metodologia STRIDE e il processo di modellazione delle minacce Microsoft. Strumento di modellazione delle minacce Microsoft (TMT). La metodologia DREAD. La metodologia LINDDUN. La metodologia Trike. La metodologia VAST. Il framework CVSS. La metodologia di valutazione del rischio OWASP.

ATTACK GRAPH: modellazione e strumenti.

APPROCCI STANDARD AL SECURITY RISK MANAGEMENT: FISMA: categorizzazione della sicurezza (FIPS-199 e NIST SP 800-60) e requisiti minimi di sicurezza (FIPS 200 e NIST 800-53); Il quadro di gestione del rischio NIST (NIST SP 800-37); Processo di valutazione del rischio NIST (NIST SP 800-30); NIST Cybersecurity Framework: core, livelli, profili.

MATERIALE DIDATTICO

Libro di testo:

Marvin Rausand, Stein Haugen. Risk assessment – Theory, Methods and Applications. Second edition Wiley.

Dispense e presentazioni fornite dal docente relative ad argomenti teorici e applicativi trattati al corso.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso prevede circa il 70% di lezioni frontali in cui vengono affrontati gli argomenti teorici, mentre il restante 30% è riservato ad esercitazioni e ad interventi seminariali da parte di esperti nello sviluppo e nella gestione di sistemi critici.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	
discussione di elaborato progettuale	x
altro	

In caso di prova scritta i quesiti sono (*)	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

La verifica dell'apprendimento prevede una prova orale e la discussione di un elaborato.

b) Modalità di valutazione:

SCHEDA DELL'INSEGNAMENTO (SI)

"INGEGNERIA DEL SUONO"

SSD ING-INF/03

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: FRANCESCO MARIA SACERDOTI
TELEFONO: +39 3355824909
EMAIL: FRANCESCOMARIA.SACERDOTI@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE): NO
MODULO (EVENTUALE): -
CANALE (EVENTUALE): -
ANNO DI CORSO (I, II, III): I/II
SEMESTRE (I, II): I
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Nessuno

OBIETTIVI FORMATIVI

Obiettivo dell'insegnamento è fornire una conoscenza approfondita del suono e della sua elaborazione per i principali ambiti applicativi d'interesse, come la creazione di contenuti multimediali, concerti e convegni dal vivo, registrazioni di musica ed effetti speciali, preparazione dell'audio per film e programmi televisivi. La conoscenza parte dall'acustica e psicoacustica, fino ad arrivare alle apparecchiature analogiche e digitali per l'elaborazione dell'audio.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di comprendere la fisiologia del sistema acustico umano e della psicoacustica e, su queste basi, di comprendere il funzionamento della percezione acustica. Inoltre, deve conoscere le basi di funzionamento delle principali apparecchiature audio per poterle utilizzare e/o progettare al meglio nei diversi ambiti applicativi d'interesse.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di saper ragionare sulle problematiche acustiche e di saper scegliere la tecnologia più adatta per la risoluzione di problemi audio.

PROGRAMMA-SYLLABUS

Audio e Acustica

Principi della diffusione delle onde audio e dell'acustica. Decibel e livello, frequenza e lunghezza d'onda, sovrapposizione, Impedenza. L'orecchio umano. Il monitoraggio dell'audio, la radiazione acustica

Psicoacustica

La percezione umana, l'orecchio e la sua fisiologia, elaborazione del segnale audio nell'uomo. Mascheratura, filtri uditivi, la non linearità dell'orecchio, percezione della fase, loudness, mel. Ricezione binaurale, pitch e timbro. Il processo mentale di percezione acustica, localizzazione dei suoni. Effetto Hall, Franssen, Cocktail Party.

Microfoni

Tipologie di microfoni, classificazione, risposta spaziale, costruzione dei microfoni. Effetto di prossimità. Risposta in frequenza. Tipi di cardioidi e microfoni PZM. Trasduttori a carbone, cristalli e ceramici. Trasduttori dinamici e a condensatore. L'alimentazione Phantom. Risoluzione della non linearità dei microfoni a condensatore. Microfoni a nastro. Misure sui microfoni: sensibilità, rumore termico, esempio di lettura di caratteristiche di un microfono. Grounding, polarità, balanced vs. unbalanced, Impedenza. Posizionamento dei microfoni. Tecniche di microfonazione: Stereo, Mid/Side, 3 to 1, microfonazione degli strumenti musicali.

Altoparlanti

Utilizzo degli altoparlanti. Costruzione. Filtri di crossover attivi e passivi. Tipologie di altoparlanti: dinamici, dome, planari, a nastro, a corno, piezo, whizzer e coassiali. Casse: costruzione e descrizione delle tecniche costruttive. Polarità ed impedenza. Specifiche. Distorsione.

Mixer

Funzioni presenti e descrizione della loro applicabilità.

Effetti audio

Processori dinamici: compressori e limitatori, noise gate ed expander, tremolo, autopanner, volume mazimizer. Processori in frequenza: equalizzatori, Wah Wah, vibrato, formant, vocoder, talk box. Processori nel dominio del tempo: echo, riverbero, phasers e flangers. Processori di pitch. Altri effetti audio.

Altre apparecchiature audio

DI Box, patch bay, synthesiser, preamplificatori microfonici.

Audio Digitale

La digitalizzazione dell'audio. Digitalizzatori. Compressione digitale. Il protocollo MIDI. Le DAW Digital Audio Workstations. I software di composizione, mixing, mastering, elaborazione audio.

MATERIALE DIDATTICO

Libro di testo:

Ballou G. "Handbook for Sound Engineers", Elsevier ISBN: 978-0-240-80969-4

Approfondimenti:
Winer E. "The Audio Expert". Routledge ISBN: 978-0-415-78883-0
Alton Everest "The Master handbook of Acoustics" McGraw Hill, ISBN 978-0-070-19897-5
Douglas S. et al. "Audio Engineering: Know It All" Elsevier ISBN: 978-1-85617-526-5
Miles Huber D. "The MIDI Manual" Elsevier ISBN: 978-0-240-80798-0

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

La didattica è erogata:
a) per l'80% con lezioni frontali;
b) per il 20% su attività laboratoriale per lo sviluppo di applicazioni audio software per comprendere al meglio le tecniche studiate.

Gli argomenti delle lezioni frontali e dei seminari sono esposti con l'ausilio di trasparenze dettagliate, messe a disposizione dello studente nel materiale didattico tramite il sito web ufficiale del docente.
Si prevede di organizzare la visita ad uno studio di registrazione di uno dei migliori ingegneri del suono italiani.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	
Altro	

L'esame si articola in una prova solo orale, consistente su domande su altrettanti problemi esposti al corso su cui si richiede un ragionamento da parte dello studente per la risoluzione di particolari problematiche di registrazione o processing audio.

SCHEDA DELL'INSEGNAMENTO (SI)

"DISTRIBUTED CONTROL AND CYBER PHYSICAL SYSTEMS DESIGN" (ALGORITMI DISTRIBUITI E PROGETTAZIONE DEI SISTEMI DI CONTROLLO SU RETE)

SSD ING-INF/04

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: SABATO MANFREDI
TELEFONO: 081-7683845
EMAIL: SABATO.MANFREDI@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO: I/II
SEMESTRE: II
CFU: 6

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Conoscenze di base sui sistemi di controllo a ciclo chiuso.

OBIETTIVI FORMATIVI

Il corso ha lo scopo di:

- fornire allo studente le competenze per l'analisi, la progettazione e il dimensionamento dei "sistemi di controllo su rete" (Networked Control Systems-NCSs) e dei "sistemi cyber-fisici" (Cyber-Physical Systems-CPSs) impiegati per il monitoraggio e il controllo dei processi distribuiti su rete;
- approfondire le tecniche di sintesi di algoritmi distribuiti, resilienti e fault-tolerant per la stima, il controllo e l'ottimizzazione su rete, applicabili ai moderni sistemi cyber-fisici presenti in ambito industriale (Smart Factory - Industria 4.0, sistemi di elaborazione distribuita, Internet of Things) e civile/sociale (Smart City, reti e infrastrutture di comunicazione);
- illustrare le metodologie introdotte attraverso esempi di progettazione integrata software/hardware di rappresentativi sistemi cyber-fisici.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il percorso formativo intende fornire agli studenti gli strumenti metodologici per l'analisi e la progettazione software/hardware dei moderni sistemi di controllo su rete e dei sistemi cyber-fisici. Lo studente deve dimostrare di avere appreso quali sono i requisiti peculiari delle componenti software e hardware dei sistemi di controllo su rete e dei sistemi cyber-fisici dedicati al monitoraggio e controllo dei principali processi industriali e civili. Lo studente dovrà inoltre dimostrare la conoscenza delle fasi principali della progettazione integrata software/hardware di un sistema cyber-fisico e della sintesi dei relativi algoritmi distribuiti di controllo, stima e ottimizzazione. Lo studente deve infine dimostrare di aver compreso il ruolo delle tecniche di validazione degli algoritmi e di valutazione delle performance di un sistema cyber-fisico mediante strumenti di simulazione.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di sapere formalizzare le specifiche di funzionamento di un sistema di controllo su rete e di un sistema cyber-fisico individuando i requisiti prestazionali del sistema di controllo e della rete, anche in termini di autonomia energetica. A partire dalle specifiche formali di rappresentativi sistemi cyber-fisici, poi, lo studente deve dimostrare di sapere sviluppare semplici algoritmi distribuiti per il monitoraggio, il controllo e l'ottimizzazione su rete, e di essere in grado di dimensionare i principali componenti hardware per la loro implementazione. Infine, lo studente dovrà mostrare la capacità di progettare i test di validazione degli algoritmi e del sistema cyber-fisico nel suo complesso avvalendosi anche dell'utilizzo di semplici simulatori.

PROGRAMMA-SYLLABUS

1. Introduzione ai sistemi di controllo su rete e ai sistemi cyber-fisici
   - 1.1 Processi complessi, distribuiti su rete e su larga scala
   - 1.2 Sistemi di controllo remoto
   - 1.3 Architetture centralizzate, decentralizzate e distribuite
   - 1.4 Algoritmi distribuiti
   - 1.5 Definizione e specifiche dei sistemi cyber-fisici e degli algoritmi distribuiti
   - 1.6 Esempi applicativi
2. Modello multi-layer dei sistemi cyber-fisici
   - 2.1 Livello "applicazione"
   - 2.2 livello "rete"
   - 2.3 livello "fisico"

2.4 Specifiche del livello applicazione, rete e fisico

3. Algoritmi distribuiti, progettazione e dimensionamento di un sistema cyber-fisico
   3.1 I sistemi multi-agente e gli algoritmi di consenso
   3.2 Progettazione del sistema di controllo a livello rete
   3.3 Sintesi di algoritmi distribuiti per il controllo di traffico, di congestione e bilanciamento del carico
   3.4 Progettazione del sistema di controllo a livello applicazione
   3.5 Sintesi di algoritmi cooperativi di stima, ottimizzazione e controllo su rete
   3.6 Autonomia energetica e "Energy Harvesting" in sistemi cyber-fisici. Algoritmi distribuiti di gestione energetica
   3.7 Algoritmi AI-based: Distributed Learning, federated Learning, Learning-based Distributed control
   3.8 Analisi di stabilità, convergenza e complessità computazionale degli algoritmi distribuiti
4. Resilienza e robustezza del sistema cyber-fisico e degli algoritmi distribuiti
   4.1 Effetti dei ritardi di comunicazione, delle perdite dati, del rumore di misura e di canale, e incertezze parametriche sulle prestazioni del sistema cyber-fisico
   4.2 Algoritmi distribuiti robusti, resilienti agli attacchi a livello di segnale e fault-tolerant
5. Algoritmi distribuiti per sistemi cyber-fisici basati su reti di sensori/sistemi embedded, reti di calcolatori, sistemi di elaborazione, flotte di droni e veicoli
   Esempi di applicazione delle metodologie introdotte al progetto integrato software/hardware di rappresentativi sistemi cyber-fisici per le Smart City e le Smart Factory (Industria 5.0)

MATERIALE DIDATTICO

- Appunti integrativi delle lezioni disponibili sul sito docenti
- S. Manfredi, "Multilayer Control of Networked Cyber-Physical Systems. Application to Monitoring, Autonomous and Robot Systems". Advances in Industrial Control, Springer, 2017
- A. Bemporad, M. Heemels, M. Vejdemo-Johansson, "Networked Control Systems", Lecture Notes in Control and Information Sciences, Springer, 2010

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzerà:
a) lezioni frontali per circa il 50% delle ore totali;
b) esercitazioni in aula mediante l'utilizzo di strumenti di simulazione e/o in laboratorio per circa il 50% delle ore totali

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

SCHEDA DELL'INSEGNAMENTO (SI)

"BIOINFORMATICA"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: DARIO RIGHELLI
TELEFONO:
EMAIL: DARIO.RIGHELLI@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III): I/II
SEMESTRE (I, II): II
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

PROGRAMMAZIONE

OBIETTIVI FORMATIVI

L'obiettivo del corso di Bioinformatica è quello di fornire agli studenti le nozioni specialistiche legate agli algoritmi per l'analisi di dati genomici e le loro eventuali applicazioni nella ricerca nella Biomedicina.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di essere in grado di descrivere processi di gestione e analisi di dati genomici sotto forma algoritmica e comprendere le metodologie adottate nell'ambito dell'elaborazione di dati molecolari.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di consultare le principali banche dati genomiche pubbliche e di descrivere e applicare procedure computazioni complesse per poter estrarre informazioni utili ai fini della ricerca biomedica.

PROGRAMMA-SYLLABUS

- Introduzione alla Biologia Molecolare e banche date biologiche
- Programmazione dinamica e allineamento di sequenze: Longest Common Subsequence, Edit Distance, Allineamento Locale, Allineamento Globale, Matrici di Sostituzione, Allineamenti Multipli.
- Modelli HMM per modeling di sequenze genomiche: Algoritmo di Viterbi, Algoritmo Forward, Posterior Decoding, Algoritmo di Baum-Welsh. Applicazioni alla classificazione di sequenze genomiche. Rappresentazione di Multiallineamenti.
- Introduzione al Next Generation Sequencing
- Algoritmi per Genome Assembly, Grafi di de Brujin, percorsi hamiltoniani e percorsi euleriani
- Algoritmi per Genome Mapping: Trie per pattern matching, Suffix Trie Tree Matching, Suffix Tree Matching, Suffix Array, Trasformata di Burrows-Wheeler e sua inversa, pattern matching con BWT
- Filogenesi molecolare e alberi filogenetici: UPGMA e Neighbor Joining
- Analisi di espressione differenziale, test statistici, analisi di arricchimento

MATERIALE DIDATTICO

- Bioinformatics Algorithms: An Active Learning Approach, by Pavel A. Pevzner and Phillip Compeau https://www.bioinformaticsalgorithms.org/
- Richard Durbin, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids
- Appunti delle lezioni

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il docente utilizzera':
a) lezioni frontali per circa il 70% delle ore totali;
b) esercitazioni per approfondire praticamente aspetti teorici per 12 ore
c) seminari per 4 ore.

La didattica verrà erogata come lezioni frontali ed esercitazioni al calcolatore.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	X
discussione di elaborato progettuale	X
altro	

SCHEDA DELL'INSEGNAMENTO (SI)

"QUANTUM INFORMATION"

SSD ING-INF/03

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO 2026-2027

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: ANGELA SARA CACCIAPUOTI
TELEFONO: 081-7683793
EMAIL: ANGELASARA.CACCIAPUOTI@UNINA.IT

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE):
MODULO (EVENTUALE):
CANALE (EVENTUALE):
ANNO DI CORSO (I, II, III):I/II
SEMESTRE (I, II):I
CFU:6

INSEGNAMENTI PROPEDEUTICI (se previsti dal Regolamento del CdS)

Nessuno

EVENTUALI PREREQUISITI

Conoscenze di base di algebra lineare

OBIETTIVI FORMATIVI

L'obiettivo del corso è fornire agli studenti una visione ampia dell'informazione e computazione quantistica da una prospettiva dell'ingegneria delle comunicazioni. Nello specifico, gli studenti familiarizzeranno con gli elementi di base della teoria dell'informazione quantistica, quali qubit, superposition, quantum measurement, no-cloning ed entanglement. Partendo da queste premesse, verranno discusse le principali applicazioni, incluse comunicazioni sicure – analizzando tecniche di Quantum Key Distribution (QKD) – e tecniche di comunicazione quantistica basate su entanglement – quali superdense coding e quantum teleportation. In tali scenari di trasmissione di informazione classica e quantistica, si forniranno inoltre agli studenti gli strumenti per comprendere le peculiarità del rumore quantistico rispetto al rumore classico. Gli studenti acquisiranno anche la capacità di comprendere le ragioni per cui l'elaborazione dell'informazione quantistica può abilitare tecniche di machine learning e artificial intelligence caratterizzate da prestazioni superiori a quelle garantite da approcci classici. Gli studenti avranno l'opportunità di eseguire semplici esperimenti su un vero computer quantistico tramite la piattaforma IBM Q-Experience.
Infine, il corso vuole fornire allo studente contenuti e linguaggio necessari per consentirgli di approfondire autonomamente le tematiche trattate nel corso, di seguire seminari di approfondimento.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Il corso si propone di fornire agli studenti i principi e gli strumenti metodologici necessari per apprendere e comprendere le problematiche che sorgono con la trasmissione e l'elaborazione dell'informazione quantistica. In particolare lo studente deve acquisire gli strumenti per comprendere le peculiarità del rumore quantistico rispetto al rumore classico. Inoltre, il corso consente agli studenti di comprendere le somiglianze e le differenze tra comunicazioni classiche e quantistiche, e di cogliere le implicazioni e le opportunità abilitate da tali differenze.

Capacità di applicare conoscenza e comprensione

Gli studenti devono dimostrare la capacità di applicare le conoscenze acquisite e gli strumenti metodologici all'analisi e alla progettazione di tecniche di comunicazione quantistica, sia in condizioni ideali che in presenza di rumore quantistico. Gli studenti devono anche essere in grado di approfondire autonomamente aspetti delle tematiche trattate nel corso e di seguire seminari di approfondimento, sfruttando i contenuti e il linguaggio forniti dal corso.

PROGRAMMA-SYLLABUS

Parte I: Fondamenti

- Informazione Quantistica: Qubit vs Bit, Spazio di Hilbert, notazione ket-bra, Sfera di Bloch, Sistemi a più qubits
    - Lab-1: Introduzione alla piattaforma IBM Quantum (IBM-Q) Experience
    - Lab-2: Visualizzazione di stati quantistici su IBM-Q
- Computazione Quantistica: Trasformazioni di stati quantistici, Theorema del No-Cloning, principali gates quantistici, la misura quantistica
    - Lab-3: Trasformazioni di stati quantistici su IBM-Q
- L'entanglement quantistico: Stati di Bell, paradosso EPR
    - Lab-4: Generazione di entanglement sulla piattaforma IBM-Q
- Rumore quantistico: stati puri e stati misti, operatore densità, decoerenza, modelli di canale quantistico

Parte II: Applicazioni

- Comunicazioni Sicure: principi di crittografia quantistica, protocollo BB84, protocollo Ekert-91, implementazioni pratiche
    - Lab-5: protocollo BB84 su IBM-Q
- Comunicazioni quantistiche: protocollo di teleporting, effetti di rumore sul teleporting, fidelity
    - Lab-6: protocollo di teleporting su IBM-Q
    - Lab-7: state tomography su IBM-Q
    - Lab-8: process tomography su IBM-Q
- Elaborazione dell'informazione quantistica per Machine Learning e Artificial Intelligence: semplici routines quantistiche, amplitude amplification e interferenza quantistica, cenni al Grover's search learning, tecniche di elaborazione quantistica distribuita
    - Lab-9: routines quantistiche su IBM-Q

MATERIALE DIDATTICO

Dispense/Slides redatte dal docente e disponibili nell'area dedicata su docenti.unina.it.

Libri di testo consigliati:

- Nielsen and Chuang, "Quantum computation and information", Cambridge University Press, 10th Edition, 2020
- Rieffel and Polak, "Quantum Computing: a Gentle Introduction", MIT Press, 2011

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso è organizzato integrando lezioni frontali con sessioni di laboratorio interattive. Durante il corso saranno inoltre organizzati seminari invitando esperti negli ambiti di interesse e saranno adottati metodi di insegnamento innovativi, come flipped classroom e feedback teaching strategies.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	
discussione di elaborato progettuale	X
altro	

COURSE DETAILS

"ROBOTICS (ROBOTICS LAB)"

SSD ING-INF/04

MASTER DEGREE PROGRAM: COMPUTER ENGINEERING

ACADEMIC YEAR 2024-2025

GENERAL INFORMATION – TEACHER REFERENCES

TEACHER: MARIO SELVAGGIO
PHONE: +39 081 76(83843)
EMAIL: MARIO.SELVAGGIO@UNINA.IT

GENERAL INFORMATION ABOUT THE COURSE

YEAR OF THE DEGREE PROGRAMME: II
SEMESTER: II
CFU: 6

REQUIRED PRELIMINARY COURSES (If mentioned in the course structure "Ordinamento")

PREREQUISITES (IF APPLICABLE)

Basic knowledge of robotics, control, machine learning, programming

LEARNING GOALS

This course will provide the fundamentals tools for building robotic applications using the Robot Operating System (ROS), that provides access to a large set of open-source software modules. Students will learn from partaking in practical activities involving real hardware and software development.

EXPECTED LEARNING OUTCOMES (DUBLIN DESCRIPTORS)

Knowledge and understanding

The course path aims to provide students with the software tools for simulating and controlling robotic systems. The Robot Operating System (ROS) will be introduced and used to control robots using basic machine learning and control schemes. The student must demonstrate knowledge in the use of software modules and in the development of machine learning and control algorithms for robot kinematics and dynamics, trajectory planning, vision-based control and mobile robots autonomous navigation.

Applying knowledge and understanding

The student must demonstrate being able to develop software modules and apply them to practical case studies concerning both open-chain robot manipulators and mobile robots. Starting from these, (s)he must demonstrate that (s)he is able to develop machine learning and sensor-based control techniques and validate them using simulated and real robots.

COURSE CONTENT/SYLLABUS

Introduction and course overview

Programming for robotics:

- Preliminary setup: linux operating system, git, docker, C++ programming
- ROS introduction, programming & tools
- ROS robot simulation
- ROS robot motion planning

Motion control of robotic manipulators:

- Kinematics and dynamic control algorithms
- Trajectory planning

Robotic vision:

- Computer vision
- Visual servoing

Control of mobile robots:

- Differential drive mobile robot
- Navigation & SLAM

READINGS/BIBLIOGRAPHY

- L. Joseph, J. Cacace, *Mastering ROS for Robotics Programming*, 2nd Edition, Packt, Birmingham, 2018, ISBN 9781801071024
- Lecture notes available at http://wpage.unina.it/mario.selvaggio/teaching.html

TEACHING METHODS

The teacher will use: a) frontal lessons for about 60% of the total hours, b) classroom exercises for about 40% of the total hours.

EXAMINATION/EVALUATION CRITERIA

a) Exam type:

Exam type	
written and oral	
only written	
only oral	X
project discussion	X
other	

In case of a written exam, questions refer to:	Multiple choice answers	
	Open answers	
	Numerical exercises	

(*) multiple options are possible

Students are admitted to the oral exam after carrying out a project concerning the simulation of the development of control algorithms for robot manipulators and mobile robots. The exam consists of a critical discussion of the paper and in ascertaining the acquisition of the concepts and contents introduced during the lessons.

b) Evaluation pattern:

The development of the project is binding for the purposes of accessing the oral exam. The project and the oral exam each contribute 50% of the final evaluation and, therefore, the development of the project is not sufficient to pass the exam.

COURSE DETAILS

"ROBOTICS (ROBOTIC SYSTEMS)"

SSD ING-INF/04

MASTER DEGREE PROGRAM: COMPUTER ENGINEERING

ACADEMIC YEAR 2024-2025

GENERAL INFORMATION – TEACHER REFERENCES

TEACHER: BRUNO SICILIANO
PHONE: 081 768 3179
EMAIL: BRUNO.SICILIANO@UNINA.IT

GENERAL INFORMATION ABOUT THE COURSE

YEAR OF THE DEGREE PROGRAMME: II
SEMESTER: I
CFU: 6

REQUIRED PRELIMINARY COURSES (If mentioned in the course structure "Ordinamento")

PREREQUISITES (IF APPLICABLE)

Basic knowledge of linear algebra and systems theory

LEARNING GOALS

The course aims to provide the fundamentals of robotic systems.

EXPECTED LEARNING OUTCOMES (DUBLIN DESCRIPTORS)

Knowledge and understanding

The course path aims to provide students with the methodological tools for modeling, planning and control of robotic systems. Robot components, kinematics, differential kinematics, statics, trajectory planning, and basic control schemes are introduced. The student must demonstrate knowledge in the derivation of models and in the validation of algorithms for trajectory planning and kinematic inversion using simulation tools.

Applying knowledge and understanding

The student must demonstrate to be able to derive kinematic and static models and know how to apply them to practical case studies concerning open-chain robot manipulators. Starting from these, (s)he must demonstrate that (s)he is able to design kinematic control schemes and know how to validate them in the Matlab / Simulink® environment.

COURSE CONTENT/SYLLABUS

- Industrial robotics and advanced robotics
- Description and principles of operation of a robot
- Direct kinematics
- Kinematic calibration
- Differential kinematics and Jacobian
- Redundancy and singularities
- Inverse kinematics algorithms
- Kineto-statics duality
- Planning of trajectories in the joint space and in the task space
- Actuators and sensors
- Control unit
- Independent joint control

READINGS/BIBLIOGRAPHY

- B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics – Modelling, Planning and Control, Springer, London, UK, 2009, DOI: 10.1007/978-1-84628-642-1. Italian translation: Robotica – Modellistica, Pianificazione e Controllo, McGraw-Hill Libri Italia, Milano, I, 2008
- B. Siciliano, Robotics Foundations I, MOOC available on the platform www.federica.eu
- B. Siciliano, Robotics Foundations II, MOOC available on the platform www.federica.eu
- Lecture notes available on course website

TEACHING METHODS

The teacher will use: a) frontal lessons for about 70% of the total hours, b) classroom exercises for about 30% of the total hours.

EXAMINATION/EVALUATION CRITERIA

a) Exam type:

Exam type	
written and oral	
only written	
only oral	X
project discussion	X
other	

In case of a written exam, questions refer to:	Multiple choice answers	
	Open answers	
	Numerical exercises	

(*) multiple options are possible

Students are admitted to the oral exam after carrying out a design project in Matlab/Simulink® concerning the simulation of inverse kinematics algorithms for robot manipulators. The exam consists of a critical discussion of the paper and in ascertaining the acquisition of the concepts and contents introduced during the lessons.

b) Evaluation pattern:

The development of the project is binding for the purposes of accessing the oral exam. The project and the oral exam each contribute 50% of the final evaluation and, therefore, the development of the project is not sufficient to pass the exam.

COURSE DETAILS

"INSTRUMENTATION AND MEASUREMENTS FOR SMART INDUSTRY"

SSD ING-INF/07

DENOMINAZIONE DEL CORSO DI STUDIO: INGEGNERIA DELLE TELECOMUNICAZIONI E DEI MEDIA DIGITALI (P49)
ANNO ACCADEMICO 2026-2027

GENERAL INFORMATION – TEACHER REFERENCES

DOCENTE: PASQUALE ARPAIA
PHONE:
EMAIL: PASQUALE.ARPAIA@UNINA.IT

GENERAL INFORMATION ABOUT THE COURSE

YEAR OF THE DEGREE PROGRAMME: I
SEMESTER: II
CFU: 9

INSEGNAMENTI PROPEDEUTICI

Non previsti

EVENTUALI PREREQUISITI

Fondamenti della misurazione, fondamenti di informatica

OBIETTIVI FORMATIVI

Il corso ha lo scopo di applicare le tecniche delle misure elettroniche ad una problematica di rilievo. I principali obiettivi formativi riguardano la capacità di specificare, concepire, progettare, implementare, testare, e qualificare hardware e firmware per microcontrollori ed un software di monitoraggio per la misura e il processo dei dati. Si insisterà altresì su multidisciplinarità e teamworking.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e saper analizzare i requisiti, progettare concettualmente e fisicamente, testare e validare un sistema di misura basato su PC (noto anche come "virtual instrument"). Deve dimostrare di avere una conoscenza approfondita delle metodologie impiegate per affrontare un problema pratico di misura ed adottare un linguaggio tecnico appropriato. Deve inoltre essere in grado di lavorare all'interno di un gruppo multidisciplinare in un ambiente che simula una piccola azienda di ricerca e sviluppo.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di gestire una commessa affidata ad un gruppo di ricerca e sviluppo, di coglierne sia gli aspetti tecnologici che economici, nonché essere in grado di gestire le risorse a disposizione per portare a termine il compito assegnato. Il percorso formativo èorientato a trasmettere le capacità, gli strumenti metodologici e gli strumenti operativi necessari a trasferire le conoscenze acquisite, con particolare riferimento al campo delle misure, verso uno scenario applicativo di matrice industriale (che in senso ampio include anche l'ambito applicativo "health 4.0") ed affrontare problematiche rilevanti su tematiche di frontiera.

PROGRAMMA

BCI: invasivi, parzialmente invasivi, e non invasivi; attivi e passivi;
endogeni ed esogeni. Prodotti e tecnologie del mercato di oggi. Principi di etica dell'interazione cervello-computer. Hardware di acquisizione dell'attività cerebrale: tecniche di neuroimaging ed elettroencefalografia, strumentazione indossabile, panoramica di mercato, prestazioni e caratterizzazione metrologica.

Physical Artificial Intelligence: generalità sull'elaborazione di segnali cerebrali per BCI, tecniche di rimozione degli artefatti, tecniche di estrazione delle caratteristiche sintetiche, approcci alla classificazione e reti neurali. Misura di potenziali evocati visivi. Misura dell'immaginazione motoria.

***Esercitazioni**: sviluppo di applicativi per microcontrollore con accensione di LED, pressione di pulsanti e utilizzo dell'interrupt. Acquisizione di segnali elettroencefalografici con sistemi di acquisizione indossabili e portabili. Elaborazione di segnali cerebrali tramite l'analisi di Fourier e della densità spettrale di potenza. Analisi temporale dei fenomeni di sincronizzazione e desincronizzazione neuronale nell'immaginazione motoria. Approccio di elaborazione e classificazione basato su FBCSP. Elaborazione online di segnali di immaginazione motoria in Matlab/Simulink.*

3. APPLICATIVO DI MISURA 4.0 BASATO SU BCI ATTIVO E ATTUATORE ROBOTICO

Principi di project management: Pianificazione dell'attività prototipale: attività e relativi workpackages, tasks, milestones, deliverables, diagramma di Gantt. Analisi di mercato.
Trasformazione delle richieste di un committente in una offerta commerciale. Stesura documentazione. Time management.

Sviluppo applicativo: analisi dei requisiti dell'applicazione. Progettazione concettuale e fisica del sistema di acquisizione ed elaborazione dei dati cerebrali. Progettazione concettuale e implementazione con i servomotori di un braccio robotico. Integrazione sistema BCI per telemonitoraggio e telecontrollo in

ambito I4.0. Debug e test. Integrazione e prove di validazione.
Stesura della documentazione finale e realizzazione della presentazione del prototipo agli stakeholders del processo.
***Esercitazioni**: sviluppo di un applicativo per microcontrollore per il controllo di servomotori ai giunti di un braccio robotico. Controllo tramite interfaccia BCI di un sistema di telemonitoraggio per applicazioni in ambito industria 4.0.*

MATERIALE DIDATTICO

M. Rüßmann, et al, "Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries". Boston Consulting Group, 2015.
A.E. Hassanien and A. A. Azar, "Brain-computer interfaces". Switzerland, Springer, 2015.
J. Wolpaw, and E. Winter Wolpaw, eds, "Brain-computer interfaces: principles and practice". OUP USA, 2012.
D. Bansal, and M. Rashima, "EEG-Based Brain-Computer Interfaces: Cognitive Analysis and Control Applications". Academic Press, 2019.
P. Arpaia, A. Esposito, L. Gargiulo, and N. Moccaldi, "Wearable Brain-Computer Interfaces: Prototyping EEG-Based Instruments for Monitoring and Control". CRC Press, 2023.
Application notes costruttori.
Manuali componenti, demo boards e applicativi.
Slide docenti.
Appunti dalle lezioni.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO-MODULO

Il corso si articolerà in lezioni teoriche per circa il 20%, esercitazioni pratiche per il 75% circa, e seminari di esperti del settore per il restante 5%. Le esercitazioni pratiche consisteranno di un progetto multidisciplinare da portare a termine in gruppo. Le slide utilizzate e tutto il materiale prodotto durante le lezioni saranno condivisi tra i vari membri del team. Il corso è tenuto da due docenti che copriranno sia gli aspetti teorici che pratici inerenti al progetto proposto. Gli studenti saranno altresì accompagnati da tutor d'aula con esperienza specifica in acquisizione dati, elaborazione dati, e programmazione dell'interfaccia software relativa al protocollo sperimentale.
Inoltre, gli studenti avranno la possibilità di interagire con un committente (di area industriale) e con un consulente (di area accademica), nonché con uno o più partner tecnologici che li supportino nell'utilizzo della strumentazione impiegata.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

Tipo esame	
scritto	
orale	X
Discussione di un elaborato progettuale	X
altro	

In caso di prova scritta i quesiti sono	**A risposta multipla**	
	A risposta libera	
	Esercizi numerici	

b) Modalità di valutazione

L'elaborato progettuale sarà monitorato durante lo svolgimento del corso e discusso al suo termine ai fini dell'esame. Alla discussione saranno affiancate anche domande di teoria volte a

sondare le conoscenze acquisite dagli allievi. L'esito finale dell'esame sarà prevalentemente determinato sulla base dell'elaborato progettuale discusso. Le domande di teoria avranno un peso pari circa al 10% del voto finale.