

# Master Privato

## Intelligenza Artificiale e Ingegneria della Conoscenza



**tech** università  
tecnologica

## Master Privato Intelligenza Artificiale e Ingegneria della Conoscenza

- » Modalità: online
- » Durata: 12 mesi
- » Titolo: TECH Università Tecnologica
- » Dedizione: 16 ore/settimana
- » Orario: a scelta
- » Esami: online

Accesso al sito web: [www.techitute.com/it/informatica/master/master-intelligenza-artificiale-ingegneria-conoscenza](http://www.techitute.com/it/informatica/master/master-intelligenza-artificiale-ingegneria-conoscenza)

# Indice

01

Presentazione

---

*pag. 4*

02

Obiettivi

---

*pag. 8*

03

Competenze

---

*pag. 14*

04

Struttura e contenuti

---

*pag. 18*

05

Metodologia

---

*pag. 30*

06

Titolo

---

*pag. 38*

# 01

# Presentazione

Questo programma è pensato per i professionisti del settore ingegneristico che vogliono immergersi nell'entusiasmante mondo dell'Intelligenza Artificiale e dell'Ingegneria della Conoscenza. Grazie a una preparazione altamente competente saranno in grado di avanzare in modo solido e solvibile in questo campo, raggiungendo le competenze personali e professionali necessarie per lavorare come un esperto del settore. Un programma completo ed efficace che ti eleverà al massimo livello di competenza.





“

*Diventa uno dei professionisti più richiesti del momento. Ottieni una qualifica di Master Privato in Intelligenza Artificiale e Ingegneria della Conoscenza grazie a questo programma molto completo”*

Gli sviluppi basati sull'intelligenza artificiale hanno raggiunto numerose applicazioni nel campo dell'ingegneria. Dall'automazione di numerose procedure nell'industria e nelle aziende, al controllo dei processi stessi. Di conseguenza, gli ingegneri professionisti devono conoscere e padroneggiare il funzionamento di queste tecniche complesse.

Questa conoscenza essenziale diventa anche il primo passo per accedere alla capacità di sviluppo di questo tipo di tecnologia.

Nel corso di questa specializzazione, viene offerto uno scenario di lavoro reale per poter valutare l'idoneità della sua applicazione nel proprio progetto, valutandone le reali indicazioni, il metodo di sviluppo e le aspettative che si possono avere rispetto ai risultati.

Grazie all'esperienza, apprenderai come sviluppare le conoscenze necessarie per progredire in questo settore di lavoro. Questo apprendimento, che richiede necessariamente esperienza, si concilia con la preparazione a distanza e l'insegnamento pratico, offrendo un'opzione unica per dare al tuo CV la spinta che stai cercando.

Questo **Master Privato in Intelligenza Artificiale e Ingegneria della Conoscenza** possiede il programma più completo e aggiornato del mercato. Le caratteristiche principali del programma sono:

- ◆ Ultima tecnologia nel software di e-learning
- ◆ Sistema di insegnamento intensamente visivo, supportato da contenuti grafici e schematici di facile assimilazione e comprensione
- ◆ Sviluppo di casi di studio presentati da esperti attivi
- ◆ Sistemi di video interattivi di ultima generazione
- ◆ Insegnamento supportato dalla pratica online
- ◆ Sistemi di aggiornamento permanente
- ◆ Apprendimento autoregolato: piena compatibilità con altre occupazioni
- ◆ Esercizi pratici per l'autovalutazione e la verifica dell'apprendimento
- ◆ Gruppi di appoggio e sinergie educative: domande agli esperti, forum di discussione e conoscenza
- ◆ Comunicazione con l'insegnante e lavoro di riflessione individuale
- ◆ Potrai accedere ai contenuti da qualsiasi dispositivo fisso o mobile con connessione a internet
- ◆ Banche di documentazione di supporto sempre disponibili, anche dopo il programma



*Unisciti all'élite grazie a questa specializzazione altamente efficace e scoprirai nuove prospettive per il tuo futuro professionale"*

“

*Un Master Privato che ti permetterà di lavorare in tutti gli ambiti dell'Intelligenza Artificiale e dell'Ingegneria della Conoscenza con la solvibilità di un professionista di alto livello"*

Il nostro personale docente è composto da professionisti in diversi settori relazionati con questa specialità. In questo modo, ci assicuriamo di raggiungere l'obiettivo di aggiornamento educativo a cui aspiriamo. Un'equipe multidisciplinare di professionisti preparati ed esperti in diversi ambienti, che svilupperanno efficacemente le conoscenze teoriche ma, soprattutto, metteranno al tuo servizio le conoscenze pratiche derivate dalla propria esperienza: una delle qualità differenziali di questa specializzazione.

Questa padronanza della materia è completata dall'efficacia del disegno metodologico. Sviluppato da un team multidisciplinare di esperti di e-learning, l'approccio integra gli ultimi progressi della tecnologia educativa. In questo modo, potrai studiare con una serie di strumenti multimediali comodi e versatili che ti daranno l'operatività di cui hai bisogno nella tua specializzazione.

La creazione di questo programma è incentrata sull'Apprendimento Basato su Problemi: un approccio che concepisce l'apprendimento come un processo eminentemente pratico. Al fine di raggiungere questo obiettivo in modalità remota viene fatto uso di la telepratica. Grazie all'aiuto di un aggiornato sistema di video interattivi potrai acquisire le conoscenze come se stessi direttamente affrontando lo scenario in fase di studio. Un concetto che permetterà di integrare e fissare l'apprendimento in modo più realistico e permanente.

*Il nostro innovativo concetto di telepratica ti darà l'opportunità di imparare attraverso un'esperienza immersiva, che ti fornirà un'integrazione più veloce e una visione molto più realistica del contenuto: "learning from an expert".*

*Grazie a una metodologia basata su tecniche di insegnamento sperimentate, questo innovativo Master Privato in Intelligenza Artificiale e Ingegneria della Conoscenza ti condurrà verso differenti approcci di insegnamento per un apprendimento dinamico ed efficace.*



# 02 Obiettivi

Il nostro obiettivo è preparare professionisti altamente qualificati per l'esperienza lavorativa. Inoltre, questo obiettivo è completato, in modo globale, dalla promozione dello sviluppo umano che pone le basi per una società migliore. Tale obiettivo si materializza aiutando i professionisti ad accedere a un livello maggiore di competenza e di controllo. Una meta che potrai considerare acquisita in pochi mesi con una preparazione ad alta intensità e precisione.



“

*Se il tuo obiettivo è orientare la tua preparazione verso nuove strade di successo e sviluppo, questo è il tuo programma: una specializzazione che punta all'eccellenza”*



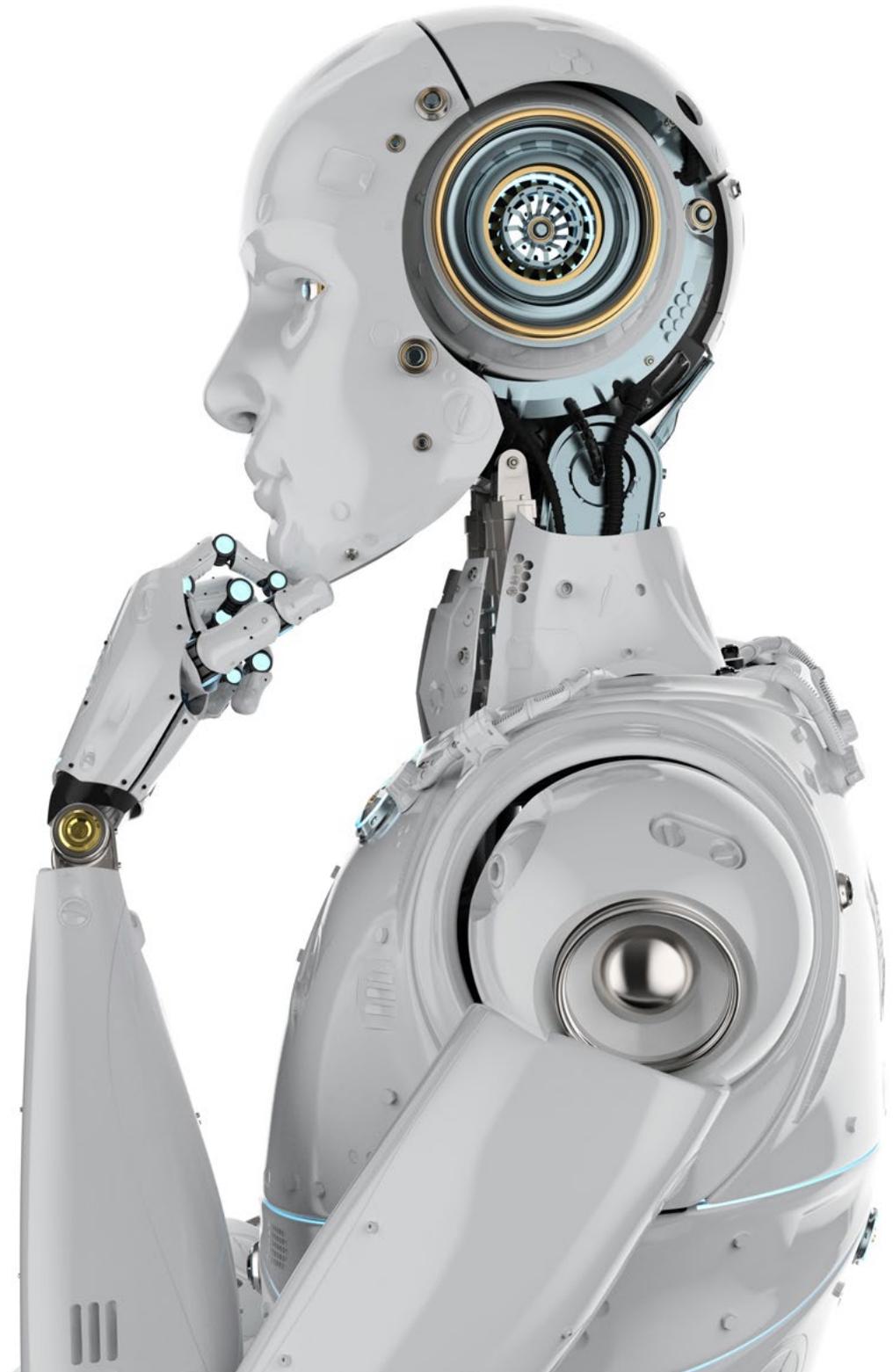
## Obiettivi generali

---

- ◆ Preparare scientificamente e tecnologicamente per l'esercizio dell'ingegneria informatica
- ◆ Ottenere un'ampia conoscenza nel campo dell'informatica
- ◆ Ottenere un'ampia conoscenza nel campo della struttura informatica
- ◆ Acquisire le conoscenze necessarie nell'ambito dell'ingegneria del software
- ◆ Rivedere le basi matematiche, statistiche e fisiche essenziali per questa materia



*Non perdere l'opportunità di aggiornarti sugli ultimi sviluppi nell'uso degli antiemorragici per integrarli nella tua pratica medica quotidiana”*





## Obiettivi specifici

---

### Modulo 1. Fondamenti di programmazione

- ◆ Comprendere la struttura di base di un computer, il software e i linguaggi di programmazione di uso generale
- ◆ Imparare a progettare e interpretare gli algoritmi, che sono la base necessaria per lo sviluppo di programmi informatici
- ◆ Comprendere gli elementi essenziali di un programma per computer, come i diversi tipi di dati, gli operatori, le espressioni, le dichiarazioni, le istruzioni di I/O e di controllo
- ◆ Comprendere le diverse strutture dati disponibili nei linguaggi di programmazione generici, sia statici che dinamici, e acquisire le conoscenze essenziali per la gestione dei file
- ◆ Comprendere le diverse tecniche di verifica dei programmi informatici e l'importanza di generare una corretta documentazione insieme a un buon codice sorgente
- ◆ Apprendere i concetti di base del linguaggio di programmazione C++, uno dei linguaggi di programmazione più utilizzati al mondo

### Modulo 2. Struttura dei dati

- ◆ Imparare i fondamenti della programmazione in linguaggio C++, tra cui classi, variabili, espressioni condizionali e oggetti
- ◆ Comprendere i tipi di dati astratti, i tipi di strutture dati lineari, le strutture dati gerarchiche semplici e complesse e la loro implementazione in C++
- ◆ Comprendere il funzionamento di strutture dati avanzate diverse da quelle abituali
- ◆ Comprendere la teoria e la pratica relative all'uso di heap e code prioritarie
- ◆ Imparare il funzionamento delle tabelle hash, come i tipi di dati astratti e le funzioni
- ◆ Comprendere la teoria dei grafi e gli algoritmi e i concetti avanzati dei grafi

### Modulo 3. Algoritmo e complessità

- ◆ Apprendere le principali strategie di progettazione degli algoritmi e i diversi metodi e le misure di calcolo di questi ultimi
- ◆ Apprendere i principali algoritmi di ordinamento utilizzati nello sviluppo del software
- ◆ Capire come funzionano i diversi algoritmi ad albero, gli *Heaps* e i grafi
- ◆ Comprendere il funzionamento degli algoritmi *Greedy*, la loro strategia e gli esempi del loro utilizzo nei principali problemi noti. Conoscere l'uso degli algoritmi *Greedy* sui grafi
- ◆ Imparare le principali strategie di ricerca del cammino minimo, con l'approccio ai problemi essenziali del campo e agli algoritmi per la loro risoluzione
- ◆ Comprendere la tecnica del backtracking e i suoi principali utilizzi, nonché altre tecniche alternative

### Modulo 4. Progettazione avanzata degli algoritmi

- ◆ Approfondire la progettazione avanzata di algoritmi, analizzando algoritmi ricorsivi e divide et impera, nonché eseguendo analisi ammortizzate
- ◆ Comprendere i concetti di programmazione dinamica e gli algoritmi per i problemi NP
- ◆ Comprendere il funzionamento dell'ottimizzazione combinatoria, nonché i diversi algoritmi di randomizzazione e gli algoritmi paralleli
- ◆ Conoscere e comprendere il funzionamento dei diversi metodi di ricerca locali e candidati
- ◆ Imparare i meccanismi della verifica formale dei programmi e di quella iterativa, tra cui la logica del primo ordine e il sistema formale di *Hoare*
- ◆ Imparare il funzionamento di alcuni dei principali metodi numerici come il metodo di bisezione, di Newton Raphson e di secante

### Modulo 5. Logica computazionale

- ◆ Apprendere i fondamenti della logica computazionale, il suo utilizzo e la sua giustificazione
- ◆ Apprendere le diverse strategie di formalizzazione e deduzione della logica proposizionale, tra cui il ragionamento naturale, la deduzione assiomatica e naturale, nonché le regole primitive del calcolo proposizionale
- ◆ Acquisire una conoscenza avanzata della logica proposizionale, approfondendo la sua semantica e le principali applicazioni di questa logica, come i circuiti logici
- ◆ Comprendere la logica dei predicati sia per il calcolo di deduzione naturale, sia per le strategie di formalizzazione e deduzione della loro logica
- ◆ Comprendere le basi del linguaggio naturale e il suo meccanismo deduttivo
- ◆ Introdurre la programmazione logica utilizzando il linguaggio PROLOG

### Modulo 6. Intelligenza Artificiale e Ingegneria della Conoscenza

- ◆ Porre le basi dell'Intelligenza Artificiale e dell'Ingegneria della Conoscenza, fornendo un breve sguardo alla storia dell'Intelligenza Artificiale fino ai giorni nostri
- ◆ Comprendere i concetti essenziali della ricerca nell'Intelligenza Artificiale, sia la ricerca informata che quella non
- ◆ Capire come funziona l'Intelligenza Artificiale nei giochi
- ◆ Apprendere i concetti fondamentali delle reti neurali e l'uso degli algoritmi genetici
- ◆ Acquisire i meccanismi appropriati per rappresentare la conoscenza, soprattutto in vista del web semantico
- ◆ Comprendere il funzionamento dei sistemi esperti e dei sistemi di supporto alle decisioni

### Modulo 7. Sistemi intelligenti

- ◆ Imparare tutti i concetti relativi alla teoria degli agenti, alla loro architettura e al processo di ragionamento
- ◆ Assimilare la teoria e la pratica alla base dei concetti di informazione e conoscenza, nonché i diversi modi di rappresentare quest'ultima
- ◆ Comprendere la teoria relativa alle ontologie e imparare i linguaggi per le ontologie e i software per la loro creazione
- ◆ Imparare diversi modelli di rappresentazione della conoscenza, come vocabolari, tassonomie, thesauri, mappe mentali e altri
- ◆ Comprendere il funzionamento dei ragionatori semantici, dei sistemi basati sulla conoscenza e dei sistemi esperti
- ◆ Conoscere il funzionamento del web semantico, il suo stato attuale e futuro, nonché le applicazioni basate sul web semantico

### Modulo 8. Apprendimento automatico e data mining

- ◆ Introdurre i processi di scoperta della conoscenza e i concetti di base dell'apprendimento automatico
- ◆ Imparare i metodi di esplorazione e pre-elaborazione dei dati, nonché i diversi algoritmi basati sugli alberi decisionali
- ◆ Comprendere il funzionamento dei metodi bayesiani, di regressione e di risposta continua
- ◆ Comprendere le diverse regole di classificazione e la valutazione dei classificatori, imparando a utilizzare le matrici di confusione e la valutazione numerica, la statistica Kappa e la curva ROC
- ◆ Acquisire le conoscenze essenziali relative al text mining, all'elaborazione del linguaggio naturale (NLP) e al *Clustering*
- ◆ Approfondire la conoscenza delle reti neurali, da quelle semplici a quelle ricorrenti



### Modulo 9. Sistemi multiagente e percezione computazionale

- ◆ Comprendere i concetti di base e avanzati relativi agli agenti e ai sistemi multi-agente
- ◆ Studiare lo standard degli agenti FIPA, prendendo in considerazione, tra le altre cose, la comunicazione tra agenti, la gestione degli agenti e l'architettura
- ◆ Approfondire l'apprendimento della piattaforma JADE (*Java Agent DEvelopment Framework*) imparando a programmare concetti sia di base che avanzati, compresi gli argomenti relativi alla comunicazione e alla scoperta degli agenti
- ◆ Porre le basi per l'elaborazione del linguaggio naturale, come il riconoscimento automatico del parlato e la linguistica computazionale
- ◆ Comprendere a fondo il funzionamento della computer vision, l'analisi delle immagini digitali, la trasformazione e la segmentazione delle immagini

### Modulo 10. Informatica bio-ispirata

- ◆ Introdurre il concetto di informatica bio-ispirata e comprendere il funzionamento dei diversi tipi di algoritmi di adattamento sociale e genetici
- ◆ Approfondire lo studio dei diversi modelli di calcolo evolutivo, conoscendone le strategie, la programmazione, gli algoritmi e i modelli basati sulla stima delle distribuzioni
- ◆ Comprendere le principali strategie di esplorazione-sfruttamento dello spazio per gli algoritmi genetici
- ◆ Comprendere il funzionamento della programmazione evolutiva applicata a problemi di apprendimento e a problemi multi-obiettivo
- ◆ Imparare i concetti essenziali relativi alle reti neurali e comprendere il funzionamento di casi d'uso reali applicati ad aree diverse come la ricerca medica, l'economia e la visione artificiale

# 03

# Competenze

Questo Master Privato in Intelligenza Artificiale e Ingegneria della Conoscenza è stato creato come strumento di alta specializzazione per il professionista. Questo programma permetterà di lavorare in tutte le aree relative all'Intelligenza Artificiale con la sicurezza di un esperto del settore.



“

*Il Master Privato in Intelligenza Artificiale e Ingegneria della Conoscenza ti fornirà le competenze personali e professionali essenziali per svolgere un ruolo adeguato in qualsiasi situazione in questo campo di intervento"*



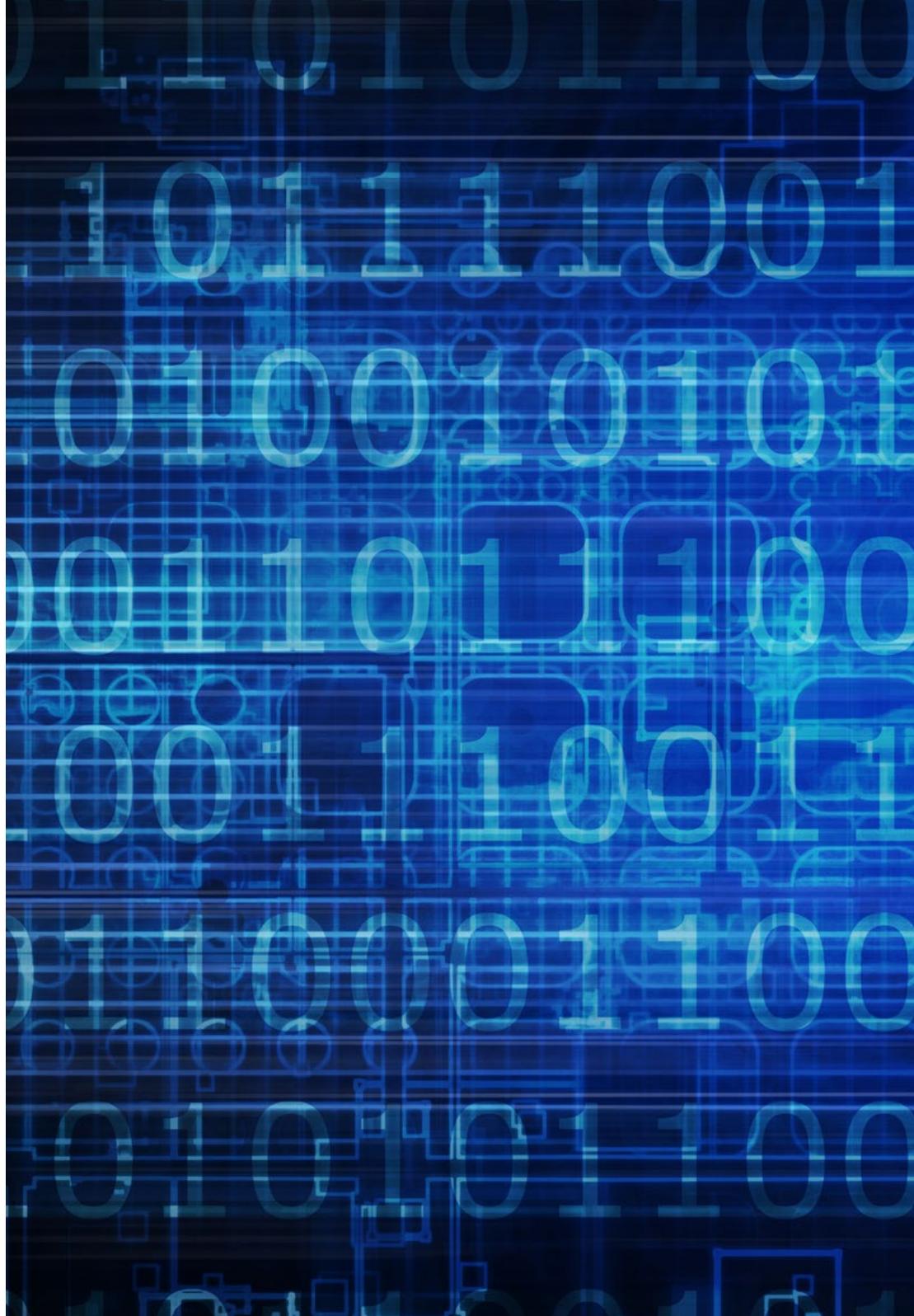
## Competenza generale

---

- ◆ Acquisire le competenze necessarie per la pratica professionale dell'ingegneria informatica con la conoscenza di tutti i fattori necessari per svolgerla con qualità e solvibilità

“

*Un'esperienza unica, cruciale  
e decisiva per potenziare la  
tua crescita professionale”*





## Competenze specifiche

---

- ◆ Sviluppare la programmazione nell'area dell'intelligenza artificiale, tenendo conto di tutti i fattori del suo sviluppo
- ◆ Conoscere la struttura dei dati della programmazione in C++
- ◆ Progettare algoritmi di base e avanzati
- ◆ Comprendere la logica computazionale e applicarla alla progettazione di progetti
- ◆ Conoscere l'Intelligenza Artificiale, i suoi utilizzi e sviluppi e implementarli ai propri progetti
- ◆ Sapere cosa sono, come funzionano e come lavorare con i sistemi intelligenti
- ◆ Padroneggiare i concetti di base dell'apprendimento automatico
- ◆ Conoscere JADE, FIPA, computer vision e altri sistemi multi-agente
- ◆ Comprendere gli algoritmi di calcolo bio-ispirati e le strategie per il loro utilizzo

# 04

## Struttura e contenuti

I contenuti di questo Master Privato sono stati sviluppati da vari settore, con con un chiaro obiettivo: permettere agli studenti di raggiungere tutte le abilità necessarie per diventare veri esperti in materia.

Un programma completo e ben strutturato che ti eleverà ai più alti standard di qualità e successo.



ARTIFICIAL  
INTELLIGENCE



PATTERN  
RECOGNITION

MA  
LEA



PROBLEM SOLVING

AUTOMATION

MACHINE LEARNING

“

*Un programma didattico molto completo, strutturato in unità didattiche ben organizzate, orientato a un apprendimento compatibile con i tuoi impegni personali e lavorativi”*

## Modulo 1. Fondamenti di programmazione

- 1.1. Introduzione alla programmazione
  - 1.1.1. Struttura di base di un computer
  - 1.1.2. Software
  - 1.1.3. Linguaggio di programmazione
  - 1.1.4. Ciclo di vita un'applicazione informatica
- 1.2. Progettazione degli algoritmi
  - 1.2.1. Risoluzione dei problemi
  - 1.2.2. Tecniche descrittive
  - 1.2.3. Elementi e struttura di un algoritmo
- 1.3. Elementi di un programma
  - 1.3.1. Origini e caratteristiche del linguaggio C++
  - 1.3.2. L'ambiente di sviluppo
  - 1.3.3. Il concetto di programma
  - 1.3.4. Tipi di dati fondamentali
  - 1.3.5. Operatori
  - 1.3.6. Espressioni
  - 1.3.7. Frasi
  - 1.3.8. Input e output di dati
- 1.4. Dichiarazioni di controllo
  - 1.4.1. Frasi
  - 1.4.2. Diramazioni
  - 1.4.3. Loop
- 1.5. Astrazione e modularità: funzioni
  - 1.5.1. Design modulare
  - 1.5.2. Concetto di funzione e utilità
  - 1.5.3. Definizione di una funzione
  - 1.5.4. Flusso di esecuzione in una chiamata di funzione
  - 1.5.5. Prototipo di una funzione
  - 1.5.6. Restituzione dei risultati
  - 1.5.7. Chiamata di una funzione: parametri
  - 1.5.8. Passaggio di parametri per riferimento e per valore
  - 1.5.9. Ambito identificatore
- 1.6. Strutture dati statiche
  - 1.6.1. Array
  - 1.6.2. Matrici: Poliedri
  - 1.6.3. Ricerca e ordinamento
  - 1.6.4. Stringhe: Funzioni di I/O per le stringhe
  - 1.6.5. Strutture: Unioni
  - 1.6.6. Nuovi tipi di dati
- 1.7. Strutture dati dinamiche: puntatori
  - 1.7.1. Concetto: Definizione di puntatore
  - 1.7.2. Operatori e operazioni con i puntatori
  - 1.7.3. Array di puntatori
  - 1.7.4. Puntatori e array
  - 1.7.5. Puntatori a stringhe
  - 1.7.6. Puntatori a strutture
  - 1.7.7. Indirizzi multipli
  - 1.7.8. Puntatori a funzioni
  - 1.7.9. Passaggio di funzioni, strutture e array come parametri di funzione
- 1.8. File
  - 1.8.1. Concetti di base
  - 1.8.2. Operazioni con i file
  - 1.8.3. Tipi di file
  - 1.8.4. Organizzazione dei file
  - 1.8.5. Introduzione ai file C++
  - 1.8.6. Gestione dei file
- 1.9. Risorse
  - 1.9.1. Definizione di risorse
  - 1.9.2. Tipi di risorse
  - 1.9.3. Vantaggi e svantaggi
  - 1.9.4. Considerazioni
  - 1.9.5. Conversione ricorsiva-iterativa
  - 1.9.6. Lo stack di ricorsione

- 1.10. Test e documentazione
  - 1.10.1. Test del programma
  - 1.10.2. Test della scatola bianca
  - 1.10.3. Test della scatola nera
  - 1.10.4. Strumenti per i test
  - 1.10.5. Documentazione del programma

## Modulo 2. Struttura dei dati

- 2.1. Introduzione alla programmazione in C++
  - 2.1.1. Classi, costruttori, metodi e attributi
  - 2.1.2. Variabili
  - 2.1.3. Espressioni condizionali e loop
  - 2.1.4. Obiettivi
- 2.2. Tipi di dati astratti (ADT)
  - 2.2.1. Tipi di dati
  - 2.2.2. Strutture di base e ADT
  - 2.2.3. Vettori e array
- 2.3. Strutture di dati lineari
  - 2.3.1. Elenco ADT: Definizione
  - 2.3.2. Elenchi collegati e doppiamente collegati
  - 2.3.3. Elenchi ordinati
  - 2.3.4. Elenchi in C++
  - 2.3.5. Stack ADT
  - 2.3.6. Coda ADT
  - 2.3.7. Stack e coda in C++
- 2.4. Strutture di dati gerarchiche
  - 2.4.1. Albero ADT
  - 2.4.2. Percorsi
  - 2.4.3. Alberi n-ari
  - 2.4.4. Alberi binari
  - 2.4.5. Alberi binari di ricerca
- 2.5. Strutture dati gerarchiche: alberi complessi
  - 2.5.1. Alberi perfettamente bilanciati o di altezza minima
  - 2.5.2. Alberi multipercorso
  - 2.5.3. Riferimenti bibliografici
- 2.6. Insieme e coda di priorità
  - 2.6.1. Insiemi di ADT
  - 2.6.2. Coda prioritaria ADT
- 2.7. Tabelle Hash
  - 2.7.1. ADT tabelle *Hash*
  - 2.7.2. Funzioni *Hash*
  - 2.7.3. Funzioni *Hash* nelle tabelle hash
  - 2.7.4. Ridispersione
  - 2.7.5. Tabelle Hash aperte
- 2.8. Grafi
  - 2.8.1. Grafi ADT
  - 2.8.2. Tipi di grafi
  - 2.8.3. Rappresentazione grafica e operazioni di base
  - 2.8.4. Progettazione dei grafi
- 2.9. Algoritmi e concetti grafici avanzati
  - 2.9.1. Problemi grafici
  - 2.9.2. Algoritmi di percorso
  - 2.9.3. Algoritmi di percorso o di ricerca
  - 2.9.4. Altri algoritmi
- 2.10. Altre strutture di dati
  - 2.10.1. Insiemi
  - 2.10.2. Array paralleli
  - 2.10.3. Tabelle dei simboli
  - 2.10.4. Tries

### Modulo 3. Algoritmo e complessità

- 3.1. Introduzione ai modelli di progettazione di algoritmi
  - 3.1.1. Risorse
  - 3.1.2. Dividi e conquista
  - 3.1.3. Altre strategie
- 3.2. Efficienza e analisi degli algoritmi
  - 3.2.1. Misure di efficienza
  - 3.2.2. Misurare l'ingresso di input
  - 3.2.3. Misurare il tempo di esecuzione
  - 3.2.4. Caso peggiore, migliore e medio
  - 3.2.5. Notazione asintotica
  - 3.2.6. Criteri di analisi matematica per algoritmi non ricorsivi
  - 3.2.7. Analisi matematica per algoritmi ricorsivi
  - 3.2.8. Analisi empirica degli algoritmi
- 3.3. Algoritmi di ordinamento
  - 3.3.1. Concetto di ordinamento
  - 3.3.2. Ordinamento delle bolle
  - 3.3.3. Ordinamento per selezione
  - 3.3.4. Ordinamento per inserimento
  - 3.3.5. Ordinamento per miscela (*Merge\_Sort*)
  - 3.3.6. Ordinamento rapido (*Quick\_Sort*)
- 3.4. Algoritmi con alberi
  - 3.4.1. Concetto di albero
  - 3.4.2. Alberi binari
  - 3.4.3. Percorsi degli alberi
  - 3.4.4. Rappresentare le espressioni
  - 3.4.5. Alberi binari ordinati
  - 3.4.6. Alberi binari bilanciati
- 3.5. Algoritmi con *Heaps*
  - 3.5.1. Gli *Heaps*
  - 3.5.2. L'algoritmo *Heapsort*
  - 3.5.3. Code prioritarie



- 3.6. Algoritmi con grafi
  - 3.6.1. Rappresentazione
  - 3.6.2. Percorso in larghezza
  - 3.6.3. Percorso in profondità
  - 3.6.4. Ordinamento topologico
- 3.7. Algoritmi *Greedy*
  - 3.7.1. La strategia *Greedy*
  - 3.7.2. Elementi della strategia *Greedy*
  - 3.7.3. Cambio valuta
  - 3.7.4. Il problema del viaggiatore
  - 3.7.5. Problema dello zaino
- 3.8. Ricerca del percorso minimo
  - 3.8.1. Il problema del percorso minimo
  - 3.8.2. Archi e cicli negativi
  - 3.8.3. Algoritmo di Dijkstra
- 3.9. Algoritmi *Greedy* sui grafi
  - 3.9.1. L'albero a sovrapposizione minima
  - 3.9.2. Algoritmo di Prim
  - 3.9.3. Algoritmo di Kruskal
  - 3.9.4. Analisi della complessità
- 3.10. *Backtracking*
  - 3.10.1. Il *Backtracking*
  - 3.10.2. Tecniche alternative
- 4.1. Analisi di algoritmi ricorsivi e divide et impera
  - 4.1.1. Porre e risolvere equazioni di ricorrenza omogenee e non
  - 4.1.2. Panoramica della strategia divide et impera
- 4.2. Analisi ammortizzata
  - 4.2.1. Analisi aggregata
  - 4.2.2. Il metodo di contabilizzazione
  - 4.2.3. Il metodo del potenziale
- 4.3. Programmazione dinamica e algoritmi per problemi NP
  - 4.3.1. Caratteristiche della programmazione dinamica
  - 4.3.2. Indietro nel tempo: backtracking
  - 4.3.3. Ramificazione e potatura
- 4.4. Ottimizzazione combinatoria
  - 4.4.1. Rappresentazione del problema
  - 4.4.2. Ottimizzazione 1D
- 4.5. Algoritmi di randomizzazione
  - 4.5.1. Esempi di algoritmi di randomizzazione
  - 4.5.2. Il teorema di Buffon
  - 4.5.3. Algoritmo di Monte Carlo
  - 4.5.4. Algoritmo di Las Vegas
- 4.6. Ricerca locale e di candidati
  - 4.6.1. Gradient *Ascent*
  - 4.6.2. Hill *Climbing*
  - 4.6.3. *Simulated Annealing*
  - 4.6.4. *Tabu Search*
  - 4.6.5. Ricerca di candidati
- 4.7. Verifica formale dei programmi
  - 4.7.1. Specifica delle astrazioni funzionali
  - 4.7.2. Il linguaggio della logica del primo ordine
  - 4.7.3. Sistema formale di Hoare
- 4.8. Verifica di programmi iterativi
  - 4.8.1. Regole del sistema formale di Hoare
  - 4.8.2. Concetto di iterazioni invariati
- 4.9. Metodi numerici
  - 4.9.1. Il metodo della bisezione
  - 4.9.2. Il metodo Newton Raphson
  - 4.9.3. Il metodo della secante
- 4.10. Algoritmi paralleli
  - 4.10.1. Operazioni binarie parallele
  - 4.10.2. Operazioni in parallelo con i grafi
  - 4.10.3. Parallelismo nel divide et impera
  - 4.10.4. Parallelismo nella programmazione dinamica

## Modulo 4. Progettazione avanzata degli algoritmi

## Modulo 5. Logica computazionale

- 5.1. Giustificazione della logica
  - 5.1.1. Oggetto di studio della logica
  - 5.1.2. A cosa serve la logica?
  - 5.1.3. Componenti e tipi di ragionamento
  - 5.1.4. Componenti di un calcolo logico
  - 5.1.5. Semantica
  - 5.1.6. Giustificazione dell'esistenza di una logica
  - 5.1.7. Come verificare che una logica sia adeguata?
- 5.2. Calcolo della deduzione naturale degli enunciati
  - 5.2.1. Linguaggio formale
  - 5.2.2. Meccanismo deduttivo
- 5.3. Strategie di formalizzazione e deduzione per la logica proposizionale
  - 5.3.1. Strategie di formalizzazione
  - 5.3.2. Ragionamento naturale
  - 5.3.3. Leggi e regole
  - 5.3.4. Deduzione assiomatica e naturale
  - 5.3.5. Il calcolo della deduzione naturale
  - 5.3.6. Regole primitive del calcolo proposizionale
- 5.4. Semantica della logica proposizionale
  - 5.4.1. Tabelle di verità
  - 5.4.2. Equivalenze
  - 5.4.3. Tautologie e contraddizioni
  - 5.4.4. Convalida di frasi proposizionali
  - 5.4.5. Convalida mediante tabelle di verità
  - 5.4.6. Convalida mediante alberi semantici
  - 5.4.7. Convalida per confutazione
- 5.5. Applicazioni della logica proposizionale: circuiti logici
  - 5.5.1. Porte di base
  - 5.5.2. Circuiti
  - 5.5.3. Modelli matematici di circuiti
  - 5.5.4. Minimizzazione
  - 5.5.5. Seconda forma canonica e forma minima nel prodotto di somme
  - 5.5.6. Altre porte

- 5.6. Calcolo deduttivo naturale dei predicati
  - 5.6.1. Linguaggio formale
  - 5.6.2. Meccanismo deduttivo
- 5.7. Strategie di formalizzazione per la logica dei predicati
  - 5.7.1. Introduzione alla formalizzazione della logica dei predicati
  - 5.7.2. Strategie di formalizzazione con quantificatori
- 5.8. Strategie di deduzione per la logica dei predicati
  - 5.8.1. Motivo dell'omissione
  - 5.8.2. Presentazione delle nuove regole
  - 5.8.3. La logica dei predicati come calcolo di deduzione naturale
- 5.9. Applicazioni della logica dei predicati: introduzione alla programmazione logica
  - 5.9.1. Presentazione informale
  - 5.9.2. Elementi di Prolog
  - 5.9.3. Rivalutazione e cut-off
- 5.10. Teoria degli insiemi, logica dei predicati e la sua semantica
  - 5.10.1. Teoria degli insiemi intuizionistica
  - 5.10.2. Introduzione alla formalizzazione della logica dei predicati

## Modulo 6. Intelligenza artificiale e Ingegneria della Conoscenza

- 6.1. Introduzione all'intelligenza artificiale e all'ingegneria della conoscenza
  - 6.1.1. Breve storia dell'intelligenza artificiale
  - 6.1.2. L'intelligenza Artificiale oggi
  - 6.1.3. Ingegneria della Conoscenza
- 6.2. Ricerca
  - 6.2.1. Concetti di ricerca comuni
  - 6.2.2. Ricerca non informata
  - 6.2.3. Ricerca informata
- 6.3. Soddisfacibilità booleana, soddisfacibilità abilità dei vincoli e pianificazione automatica
  - 6.3.1. Soddisfacibilità booleana
  - 6.3.2. Problemi di soddisfazione dei vincoli
  - 6.3.3. Pianificazione automatica e PDDL
  - 6.3.4. Pianificazione come ricerca euristica
  - 6.3.5. Pianificazione con il SAT

- 6.4. Intelligenza artificiale nei giochi
  - 6.4.1. Teoria dei giochi
  - 6.4.2. Potenziamiento Minimax e Alfa-Beta
  - 6.4.3. Simulazione: Monte Carlo
- 6.5. Apprendimento supervisionato e non
  - 6.5.1. Introduzione all'apprendimento automatico
  - 6.5.2. Classificazione
  - 6.5.3. Regressione
  - 6.5.4. Convalida dei risultati
  - 6.5.5. Raggruppamento (*Clustering*)
- 6.6. Reti neurali
  - 6.6.1. Basi biologiche
  - 6.6.2. Modello computazionale
  - 6.6.3. Reti neurali supervisionate e non
  - 6.6.4. Perceptron semplice
  - 6.6.5. Perceptron multistrato
- 6.7. Algoritmi genetici
  - 6.7.1. Storia
  - 6.7.2. Base biologica
  - 6.7.3. Codifica dei problemi
  - 6.7.4. Generazione della popolazione iniziale
  - 6.7.5. Algoritmo principale e operatori genetici
  - 6.7.6. Valutazione degli individui: *Fitness*
- 6.8. Thesauri, vocabolari, tassonomie
  - 6.8.1. Vocabolari
  - 6.8.2. Tassonomie
  - 6.8.3. Thesauri
  - 6.8.4. Ontologie
- 6.9. Rappresentazione della conoscenza: web semantico
  - 6.9.1. Web semantico
  - 6.9.2. Specifiche: RDF, RDFS e OWL
  - 6.9.3. Inferenza/ragionamento
  - 6.9.4. Linked Data

- 6.10. Sistemi esperti e DSS
  - 6.10.1. Sistemi esperti
  - 6.10.2. Sistemi di supporto decisionale

## Modulo 7. Sistemi intelligenti

- 7.1. Teoria degli agenti
  - 7.1.1. Storia del concetto
  - 7.1.2. Definizione di agente
  - 7.1.3. Agenti nell'intelligenza artificiale
  - 7.1.4. Agenti nell'Ingegneria dei software
- 7.2. Architetture di agenti
  - 7.2.1. Il processo di ragionamento dell'agente
  - 7.2.2. Agenti reattivi
  - 7.2.3. Agenti deduttivi
  - 7.2.4. Agenti ibridi
  - 7.2.5. Confronto
- 7.3. Informazione e conoscenza
  - 7.3.1. Distinzione tra dati, informazioni e conoscenza
  - 7.3.2. Valutazione della qualità dei dati
  - 7.3.3. Metodi di raccolta dei dati
  - 7.3.4. Metodi di acquisizione dei dati
  - 7.3.5. Metodi di acquisizione della conoscenza
- 7.4. Rappresentazione della conoscenza
  - 7.4.1. L'importanza della rappresentazione della conoscenza
  - 7.4.2. Definire la rappresentazione della conoscenza attraverso i suoi ruoli
  - 7.4.3. Caratteristiche di una rappresentazione della conoscenza
- 7.5. Ontologie
  - 7.5.1. Introduzione ai metadati
  - 7.5.2. Concetto filosofico di ontologia
  - 7.5.3. Concetto informatico di ontologia
  - 7.5.4. Ontologie di dominio e di livello superiore
  - 7.5.5. Come costruire un'ontologia?

- 7.6. Linguaggi ontologici e software per la creazione di ontologie
  - 7.6.1. Triple RDF, Turtle e N3
  - 7.6.2. Schema RDF
  - 7.6.3. OWL
  - 7.6.4. SPARQL
  - 7.6.5. Introduzione ai diversi strumenti per la creazione di ontologie
  - 7.6.6. Installazione e utilizzo di Protégé
- 7.7. Web semantico
  - 7.7.1. Lo stato attuale e il futuro del web semantico
  - 7.7.2. Applicazioni del web semantico
- 7.8. Altri modelli di rappresentazione della conoscenza
  - 7.8.1. Vocabolari
  - 7.8.2. Panoramica
  - 7.8.3. Tassonomie
  - 7.8.4. Thesauri
  - 7.8.5. Folksonomie
  - 7.8.6. Confronto
  - 7.8.7. Mappe mentali
- 7.9. Valutazione e integrazione delle rappresentazioni della conoscenza
  - 7.9.1. Logica dell'ordine zero
  - 7.9.2. Logico di primo ordine
  - 7.9.3. Logica descrittiva
  - 7.9.4. Relazione tra i diversi tipi di logica
  - 7.9.5. Prolog: programmazione basata sulla logica del primo ordine
- 7.10. Ragonatori semantici, sistemi basati sulla conoscenza e sistemi esperti
  - 7.10.1. Concetto di ragionatore
  - 7.10.2. Applicazioni di un ragionatore
  - 7.10.3. Sistemi basati sulla conoscenza
  - 7.10.4. MYCIN, storia dei sistemi esperti
  - 7.10.5. Elementi e architettura dei sistemi esperti
  - 7.10.6. Creazione di sistemi esperti

## Modulo 8. Apprendimento automatico e data mining

- 8.1. Introduzione ai processi di scoperta della conoscenza e ai concetti di base dell'apprendimento automatico
  - 8.1.1. Concetti chiave dei processi di scoperta della conoscenza
  - 8.1.2. Prospettiva storica sui processi di scoperta della conoscenza
  - 8.1.3. Fasi dei processi di scoperta della conoscenza
  - 8.1.4. Tecniche utilizzate nei processi di scoperta della conoscenza
  - 8.1.5. Caratteristiche dei buoni modelli di apprendimento automatico
  - 8.1.6. Tipi di informazioni sull'apprendimento automatico
  - 8.1.7. Concetti di base dell'apprendimento
  - 8.1.8. Concetti di base dell'apprendimento non supervisionato
- 8.2. Analisi e pre-elaborazione dei dati
  - 8.2.1. Trattamento dei dati
  - 8.2.2. Trattamento dei dati nel flusso di analisi dei dati
  - 8.2.3. Tipi di dati
  - 8.2.4. Trasformazione dei dati
  - 8.2.5. Visualizzazione ed esplorazione di variabili continue
  - 8.2.6. Visualizzazione ed esplorazione di variabili categoriche
  - 8.2.7. Misure di correlazione
  - 8.2.8. Rappresentazioni grafiche più comuni
  - 8.2.9. Introduzione all'analisi multivariata e alla riduzione delle dimensioni
- 8.3. Alberi decisionali
  - 8.3.1. Algoritmo ID3
  - 8.3.2. Algoritmo C4.5
  - 8.3.3. Sovrallenamento e potatura
  - 8.3.4. Analisi dei risultati
- 8.4. Valutazione dei classificatori
  - 8.4.1. Matrici di confusione
  - 8.4.2. Matrici di valutazione numerica
  - 8.4.3. Statistica Kappa
  - 8.4.5. La curva ROC

- 8.5. Regole di classificazione
  - 8.5.1. Misure di valutazione delle regole
  - 8.5.2. Introduzione alla rappresentazione grafica
  - 8.5.3. Algoritmo di sovrapposizione sequenziale
- 8.6. Reti neurali
  - 8.6.1. Concetti di base
  - 8.6.2. Reti neurali semplici
  - 8.6.3. Algoritmo di *Backpropagation*
  - 8.6.4. Introduzione alle reti neurali ricorrenti
- 8.7. Metodi bayesiani
  - 8.7.1. Concetti di base della probabilità
  - 8.7.2. Teorema di Bayes
  - 8.7.3. Naive Bayes
  - 8.7.4. Introduzione alle reti bayesiane
- 8.8. Modelli di regressione e di risposta continua
  - 8.8.1. Regressione lineare semplice
  - 8.8.2. Regressione lineare multipla
  - 8.8.3. Regressione logistica
  - 8.8.4. Alberi di regressione
  - 8.8.5. Introduzione alle macchine a vettori di supporto (SVM)
  - 8.8.6. Misure di bontà di adattamento
- 8.9. *Clustering*
  - 8.9.1. Concetti di base
  - 8.9.2. Clustering gerarchico
  - 8.9.3. Metodi probabilistici
  - 8.9.4. Algoritmo EM
  - 8.9.5. Metodo B-Cubed
  - 8.9.6. Metodi impliciti
- 8.10. Estrazione di testi ed elaborazione del linguaggio naturale (NLP)
  - 8.10.1. Concetti di base
  - 8.10.2. Creazione del corpus
  - 8.10.3. Analisi descrittiva
  - 8.10.4. Introduzione alla sentiment analysis

## Modulo 9. Sistemi multiagente e percezione computazionale

- 9.1. Agenti e sistemi multiagente
  - 9.1.1. Concetto di agente
  - 9.1.2. Architettura
  - 9.1.3. Comunicazione e coordinamento
  - 9.1.4. Linguaggi e strumenti di programmazione
  - 9.1.5. Applicazioni degli agenti
  - 9.1.6. La FIPA
- 9.2. Lo standard dell'agente: FIPA
  - 9.2.1. Comunicazione tra gli agenti
  - 9.2.2. La gestione degli agenti
  - 9.2.3. Architettura astratta
  - 9.2.4. Altre specifiche
- 9.3. La piattaforma JADE
  - 9.3.1. Agenti software secondo JADE
  - 9.3.2. Architettura
  - 9.3.3. Installazione ed esecuzione
  - 9.3.4. Pacchetti JADE
- 9.4. Programmazione di base con JADE
  - 9.4.1. La console di gestione
  - 9.4.2. Creazione di agenti di base
- 9.5. Programmazione avanzata con JADE
  - 9.5.1. Creazione avanzata di agenti
  - 9.5.2. Comunicazione tra gli agenti
  - 9.5.3. Scoperta dell'agente
- 9.6. Visione artificiale
  - 9.6.1. Elaborazione e analisi delle immagini digitali
  - 9.6.2. Analisi delle immagini e visione artificiale
  - 9.6.3. Elaborazione delle immagini e visione umana
  - 9.6.4. Sistema di acquisizione delle immagini
  - 9.6.5. Creazione dell'immagine e percezione

- 9.7. Analisi delle immagini digitali
  - 9.7.1. Fasi del processo di analisi delle immagini
  - 9.7.2. Pre-elaborazione
  - 9.7.3. Operazioni di base
  - 9.7.4. Filtraggio spaziale
- 9.8. Trasformazione e segmentazione delle immagini digitali
  - 9.8.1. Trasformate di Fourier
  - 9.8.2. Filtraggio di frequenza
  - 9.8.3. Concetti di base
  - 9.8.4. Soglia
  - 9.8.5. Rilevamento dei contorni
- 9.9. Riconoscimento della forma
  - 9.9.1. Estrazione delle caratteristiche
  - 9.9.2. Algoritmi di classificazione
- 9.10. Elaborazione di linguaggio naturale
  - 9.10.1. Riconoscimento vocale automatico
  - 9.10.2. Linguistica computazionale

## Modulo 10. Informatica bio-ispirata

- 10.1. Introduzione all'informatica bio-ispirata
  - 10.1.1. Introduzione all'informatica bio-ispirata
- 10.2. Algoritmi di adattamento sociale
  - 10.2.1. Calcolo bio-ispirato basato su colonie di formiche
  - 10.2.2. Varianti degli algoritmi di colonia di formiche
  - 10.2.3. Elaborazione particellare basata su cloud
- 10.3. Algoritmi genetici
  - 10.3.1. Struttura generale
  - 10.3.2. Implementazioni dei principali operatori

- 10.4. Strategie spaziali di esplorazione-sfruttamento per algoritmi genetici
  - 10.4.1. Algoritmo CHC
  - 10.4.2. Problemi multimodali
- 10.5. Modelli di calcolo evolutivo (I)
  - 10.5.1. Strategie evolutive
  - 10.5.2. Programmazione evolutiva
  - 10.5.3. Algoritmi basati sull'evoluzione differenziale
- 10.6. Modelli di calcolo evolutivo (II)
  - 10.6.1. Modelli evolutivi basati sulla stima delle distribuzioni (EDA)
  - 10.6.2. Programmazione genetica
- 10.7. Programmazione evolutiva applicata ai problemi di apprendimento
  - 10.7.1. Apprendimento basato sulle regole
  - 10.7.2. Metodi evolutivi nei problemi di selezione delle istanze
- 10.8. Problemi multi-obiettivo
  - 10.8.1. Concetto di dominanza
  - 10.8.2. Applicazione degli algoritmi evolutivi ai problemi multi-obiettivo
- 10.9. Reti neurali (I)
  - 10.9.1. Introduzione alle reti neurali
  - 10.9.2. Esempio pratico con le reti neurali
- 10.10. Reti neurali (II)
  - 10.10.1. Casi di utilizzo delle reti neurali nella ricerca medica
  - 10.10.2. Casi di utilizzo delle reti neurali in economia
  - 10.10.3. Casi di utilizzo delle reti neurali nella visione artificiale

Solution Explorer

Search Solution Explorer (Ctrl+)

- AboutWindow.cpp
- auth.c
- auth.h
- \_auth.h
- get\_access\_token.c
- client.c
- client.h
- \_client.h
- client\_block(int, int)
- client\_chunk(int, int)
- client\_connect(char)
- client\_disable()
- client\_enable()
- client\_light(int, int, i)
- client\_login(const cl)
- client\_position(float)
- client\_recv()
- client\_send(char \*)
- client\_sign(int, int, i)
- client\_start()
- client\_stop()
- client\_talk(const cha)
- client\_version(int)
- DEFAULT\_PORT
- get\_client\_enabled()
- config.h
- ConnectionDialog.cpp
- \_make\_sphere(float
- .../float

ConsoleApplication1

```

        continue;
    }
    float du = (tiles[i] % 16) * s;
    float dv = (tiles[i] / 16) * s;
    int flip = ao[i][0] + ao[i][3] > ao[i][1] + ao[i][2];
    for (int v = 0; v < 6; v++) {
        int j = flip ? flipped[i][v] : indices[i][v];
        *(d++) = x + n * positions[i][j][0];
        *(d++) = y + n * positions[i][j][1];
        *(d++) = z + n * positions[i][j][2];
        *(d++) = normals[i][0];
        *(d++) = normals[i][1];
        *(d++) = normals[i][2];
        *(d++) = du + (uvs[i][j][0] ? b : a);
        *(d++) = dv + (uvs[i][j][1] ? b : a);
        *(d++) = ao[i][j];
        *(d++) = light[i][j];
    }
}
}
}

```

```

void make_cube(
    float *data, float ao[6][4], float light[6][4],
    int left, int right, int top, int bottom, int front, int back,
    float x, float y, float z, float n, int w)
{
    int wleft = blocks[w][0];
    int wright = blocks[w][1];
    int wtop = blocks[w][2];
    int wbottom = blocks[w][3];
    int wfront = blocks[w][4];
    int wback = blocks[w][5];
    make_cube_faces(
        data, ao, light,
        left, right, top, bottom, front, back,
        wleft, wright, wtop, wbottom, wfront, wback,
        x, y, z, n);
}

```

# 05 Metodologia

Questo programma ti offre un modo differente di imparare. La nostra metodologia si sviluppa in una modalità di apprendimento ciclico: ***il Relearning***.

Questo sistema di insegnamento viene applicato nelle più prestigiose facoltà di medicina del mondo ed è considerato uno dei più efficaci da importanti pubblicazioni come il ***New England Journal of Medicine***.



“

*Scopri il Relearning, un sistema che abbandona l'apprendimento lineare convenzionale, per guidarti attraverso dei sistemi di insegnamento ciclici: una modalità di apprendimento che ha dimostrato la sua enorme efficacia, soprattutto nelle materie che richiedono la memorizzazione”*

## Caso di Studio per contestualizzare tutti i contenuti

Il nostro programma offre un metodo rivoluzionario per sviluppare le abilità e le conoscenze. Il nostro obiettivo è quello di rafforzare le competenze in un contesto mutevole, competitivo e altamente esigente.

“

*Con TECH potrai sperimentare un modo di imparare che sta scuotendo le fondamenta delle università tradizionali in tutto il mondo”*



*Avrai accesso a un sistema di apprendimento basato sulla ripetizione, con un insegnamento naturale e progressivo durante tutto il programma.*



*Imparerai, attraverso attività collaborative e casi reali, la risoluzione di situazioni complesse in ambienti aziendali reali.*

## Un metodo di apprendimento innovativo e differente

Questo programma di TECH consiste in un insegnamento intensivo, creato ex novo, che propone le sfide e le decisioni più impegnative in questo campo, sia a livello nazionale che internazionale. Grazie a questa metodologia, la crescita personale e professionale viene potenziata, effettuando un passo decisivo verso il successo. Il metodo casistico, la tecnica che sta alla base di questi contenuti, garantisce il rispetto della realtà economica, sociale e professionale più attuali.

“

*Il nostro programma ti prepara ad affrontare nuove sfide in ambienti incerti e a raggiungere il successo nella tua carriera”*

Il Metodo Casistico è stato il sistema di apprendimento più usato nelle migliori Scuole di Informatica del mondo da quando esistono. Sviluppato nel 1912 affinché gli studenti di Diritto non imparassero la legge solo sulla base del contenuto teorico, il metodo casistico consisteva nel presentare loro situazioni reali e complesse per prendere decisioni informate e giudizi di valore su come risolverle. Nel 1924 fu stabilito come metodo di insegnamento standard ad Harvard.

Cosa dovrebbe fare un professionista per affrontare una determinata situazione?

Questa è la domanda con cui ti confrontiamo nel metodo dei casi, un metodo di apprendimento orientato all'azione. Durante il corso, gli studenti si confronteranno con diversi casi di vita reale. Dovranno integrare tutte le loro conoscenze, effettuare ricerche, argomentare e difendere le proprie idee e decisioni.

## Metodologia Relearning

TECH coniuga efficacemente la metodologia del Caso di Studio con un sistema di apprendimento 100% online basato sulla ripetizione, che combina diversi elementi didattici in ogni lezione.

Potenziamo il Caso di Studio con il miglior metodo di insegnamento 100% online: il Relearning.

*Nel 2019 abbiamo ottenuto i migliori risultati di apprendimento di tutte le università online del mondo.*

In TECH imparerai con una metodologia all'avanguardia progettata per formare i manager del futuro. Questo metodo, all'avanguardia della pedagogia mondiale, si chiama Relearning.

La nostra università è l'unica autorizzata a utilizzare questo metodo di successo. Nel 2019, siamo riusciti a migliorare il livello di soddisfazione generale dei nostri studenti (qualità dell'insegnamento, qualità dei materiali, struttura del corso, obiettivi...) rispetto agli indicatori della migliore università online.



Nel nostro programma, l'apprendimento non è un processo lineare, ma avviene in una spirale (impariamo, disimpariamo, dimentichiamo e re-impariamo). Pertanto, combiniamo ciascuno di questi elementi in modo concentrico. Questa metodologia ha formato più di 650.000 laureati con un successo senza precedenti in campi diversi come la biochimica, la genetica, la chirurgia, il diritto internazionale, le competenze manageriali, le scienze sportive, la filosofia, il diritto, l'ingegneria, il giornalismo, la storia, i mercati e gli strumenti finanziari. Tutto questo in un ambiente molto esigente, con un corpo di studenti universitari con un alto profilo socio-economico e un'età media di 43,5 anni.

*Il Relearning ti permetterà di apprendere con meno sforzo e più performance, impegnandoti maggiormente nella tua specializzazione, sviluppando uno spirito critico, difendendo gli argomenti e contrastando le opinioni: un'equazione diretta al successo.*

Dalle ultime evidenze scientifiche nel campo delle neuroscienze, non solo sappiamo come organizzare le informazioni, le idee, le immagini e i ricordi, ma sappiamo che il luogo e il contesto in cui abbiamo imparato qualcosa è fondamentale per la nostra capacità di ricordarlo e immagazzinarlo nell'ippocampo, per conservarlo nella nostra memoria a lungo termine.

In questo modo, e in quello che si chiama Neurocognitive Context-dependent E-learning, i diversi elementi del nostro programma sono collegati al contesto in cui il partecipante sviluppa la sua pratica professionale.



Questo programma offre i migliori materiali didattici, preparati appositamente per i professionisti:



#### Materiali di studio

Tutti i contenuti didattici sono creati appositamente per il corso dagli specialisti che lo impartiranno, per fare in modo che lo sviluppo didattico sia davvero specifico e concreto.

Questi contenuti sono poi applicati al formato audiovisivo che supporterà la modalità di lavoro online di TECH. Tutto questo, con le ultime tecniche che offrono componenti di alta qualità in ognuno dei materiali che vengono messi a disposizione dello studente.



#### Master class

Esistono evidenze scientifiche sull'utilità dell'osservazione di esperti terzi.

Imparare da un esperto rafforza la conoscenza e la memoria, costruisce la fiducia nelle nostre future decisioni difficili.



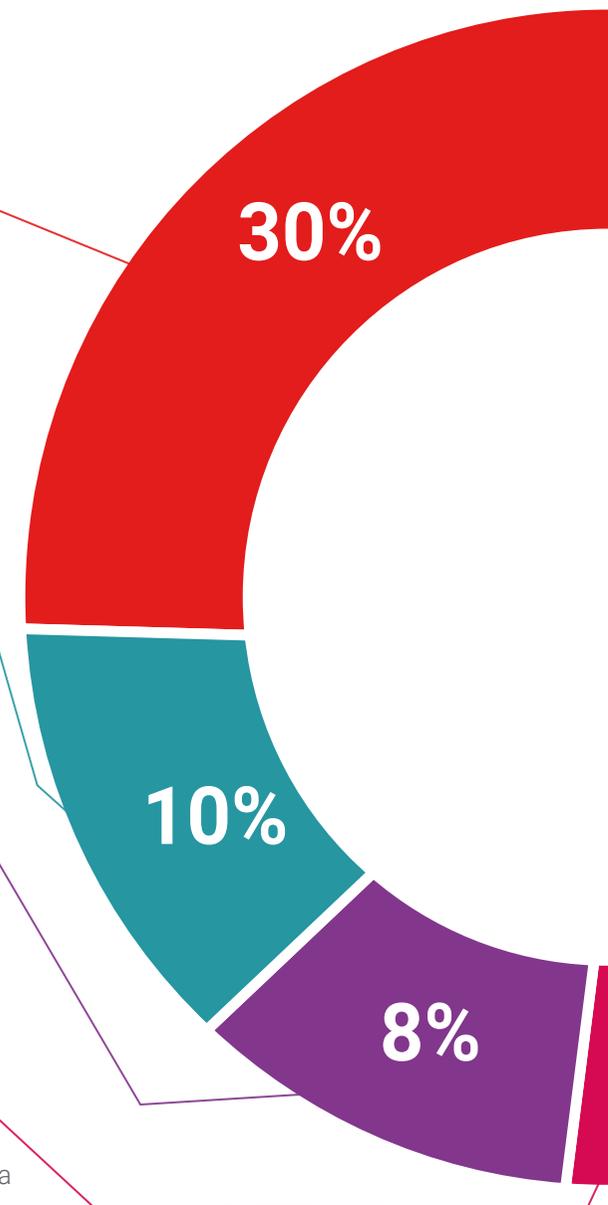
#### Pratiche di competenze e competenze

Svolgerai attività per sviluppare competenze e capacità specifiche in ogni area tematica. Pratiche e dinamiche per acquisire e sviluppare le competenze e le abilità che uno specialista deve sviluppare nel quadro della globalizzazione in cui viviamo.



#### Letture complementari

Articoli recenti, documenti di consenso e linee guida internazionali, tra gli altri. Nella biblioteca virtuale di TECH potrai accedere a tutto il materiale necessario per completare la tua specializzazione.





#### Casi di Studio

Completerai una selezione dei migliori casi di studio scelti appositamente per questo corso. Casi presentati, analizzati e monitorati dai migliori specialisti del panorama internazionale.



#### Riepiloghi interattivi

Il team di TECH presenta i contenuti in modo accattivante e dinamico in pillole multimediali che includono audio, video, immagini, diagrammi e mappe concettuali per consolidare la conoscenza.

Questo esclusivo sistema di specializzazione per la presentazione di contenuti multimediali è stato premiato da Microsoft come "Caso di successo in Europa".



#### Testing & Retesting

Valutiamo e rivalutiamo periodicamente le tue conoscenze durante tutto il programma con attività ed esercizi di valutazione e autovalutazione, affinché tu possa verificare come raggiungi progressivamente i tuoi obiettivi.



# 06 Titolo

Il Master Privato in Intelligenza Artificiale e Ingegneria della Conoscenza ti garantisce, oltre alla preparazione più rigorosa e aggiornata, l'accesso a una qualifica di Master Specialistico rilasciata da TECH Università Tecnologica.



“

*Porta a termine questo programma e ricevi la tua qualifica universitaria senza spostamenti o fastidiose formalità”*

Questo **Master Privato in Intelligenza Artificiale e Ingegneria della Conoscenza** possiede il programma più completo e aggiornato del mercato.

Dopo aver superato la valutazione, lo studente riceverà mediante lettera certificata\* con ricevuta di ritorno, la sua corrispondente qualifica di **Master Privato** rilasciata da **TECH Università Tecnologica**.

Il titolo rilasciato da **TECH Università Tecnologica** esprime la qualifica ottenuta nel Master Privato, e riunisce tutti i requisiti comunemente richiesti da borse di lavoro, concorsi e commissioni di valutazione di carriere professionali.

Titolo: **Master Privato in Intelligenza Artificiale e Ingegneria della Conoscenza**  
N. Ore Ufficiali: **1.500 O.**



\*Se lo studente dovesse richiedere che il suo diploma cartaceo sia provvisto di Apostille dell'Aia, TECH EDUCATION effettuer  le gestioni opportune per ottenerla pagando un costo aggiuntivo.

futuro  
salute fiducia persone  
educazione informazione tutor  
garanzia accreditamento insegnamento  
istituzioni tecnologia apprendimento  
comunità impegno  
attenzione personalizzata  
conoscenza presente qualità  
formazione online  
sviluppo istituzioni  
classe virtuale lingua

**tech** università  
tecnologica

**Master Privato**  
Intelligenza Artificiale  
e Ingegneria della Conoscenza

- » Modalità: **online**
- » Durata: **12 mesi**
- » Titolo: **TECH Università Tecnologica**
- » Dedizione: **16 ore/settimana**
- » Orario: **a scelta**
- » Esami: **online**

# Master Privato

## Intelligenza Artificiale e Ingegneria della Conoscenza