

# Master Privato

Creazione di Interfacce  
e Applicazioni in Rete



## Master Privato Creazione di Interfacce e Applicazioni in Rete

- » 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-creazione-interfacce-applicazioni-rete](http://www.techitute.com/it/informatica/master/master-creazione-interfacce-applicazioni-rete)

# Indice

01

Presentazione

---

*pag. 4*

02

Obiettivi

---

*pag. 8*

03

Competenze

---

*pag. 14*

04

Struttura e contenuti

---

*pag. 18*

05

Metodologia

---

*pag. 32*

06

Titolo

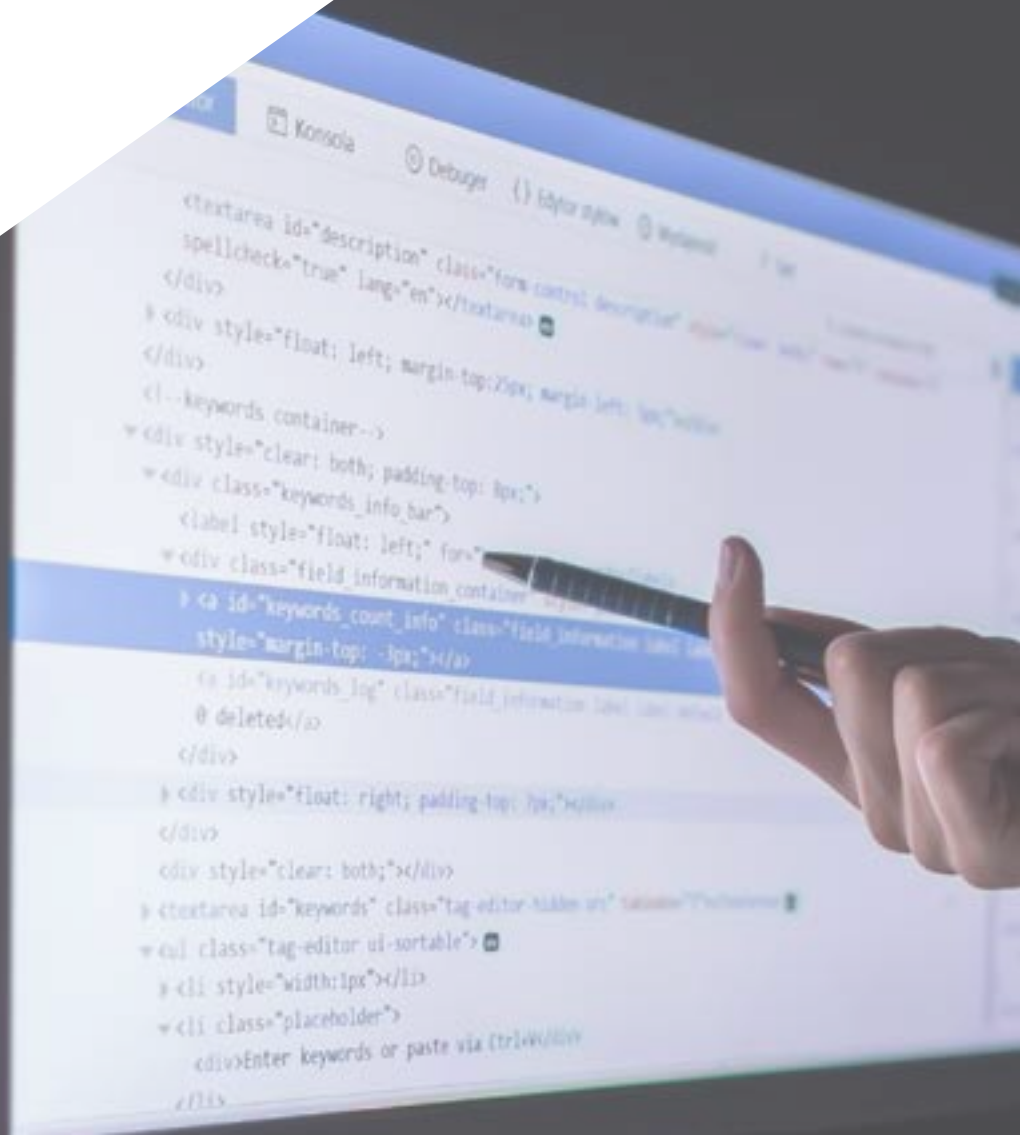
---

*pag. 40*

# 01

# Presentazione

Intervenire in modo competitivo nel campo dell'informatica richiede una solida conoscenza che includa le innovazioni e gli aggiornamenti più rilevanti nell'ingegneria del software, nelle basi matematiche, nella statistica e in altre aree. In questo programma abbiamo costruito un percorso intensivo di specializzazione attraverso il quale il professionista potrà prepararsi scientificamente e tecnologicamente nell'ingegneria informatica, con qualità, sicurezza e un obiettivo finale di eccellenza.



“

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

Questa specializzazione svilupperà i concetti necessari per lavorare nella creazione di interfacce, con la sicurezza che la padronanza di tutte le aree di conoscenza trasversali a questa materia forniscono allo specialista. Nel corso del programma verranno proposti approcci didattici innovativi per acquisire una conoscenza approfondita dell'architettura di un'applicazione distribuita. Saranno presentati gli aspetti particolari dell'architettura client-server e saranno affrontati i fondamenti e gli sviluppi essenziali dei linguaggi di programmazione più comunemente utilizzati, differenziando i linguaggi tra molte altre questioni fondamentali per il professionista.

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 progetto, valutandone le reali indicazioni, il metodo di sviluppo e le aspettative che si possono avere rispetto ai risultati.

Grazie all'esperienza, lo studente apprenderà come sviluppare le conoscenze necessarie per progredire in questo settore di lavoro. Questo programma, che richiede necessariamente esperienza, si concilia con la didattica a distanza e l'insegnamento pratico, offrendo un'opzione unica per dare al CV il valore aggiunto desiderato.



*Diventa uno dei professionisti più richiesti del momento: specializzati in ingegneria informatica attraverso il più completo e aggiornato Master Privato in Creazione di Interfacce e Applicazioni in Rete"*

Puesto **Master Privato in Creazione di Interfacce e Applicazioni in Rete** 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
- ◆ Accesso ai contenuti da qualsiasi dispositivo fisso o mobile con connessione a internet
- ◆ Banche di documentazione di supporto sempre disponibili, anche dopo il programma

“

*Grazie a una metodologia basata su tecniche di insegnamento sperimentate, questo innovativo Master Privato in Creazione di Interfacce e Applicazioni in Rete ti condurrà verso differenti approcci di insegnamento per un apprendimento dinamico ed efficace”*

Il nostro personale docente è composto da professionisti in diversi settori relazionati con questa specialità. In questo modo TECH si assicura di offrire ai propri studenti le conoscenze specialistiche necessarie. Un'equipe multidisciplinare di professionisti specializzati e con esperienza in diversi ambiti, che svilupperanno efficacemente le conoscenze teoriche e, soprattutto, metteranno al tuo servizio le conoscenze pratiche derivate dalla propria esperienza: una delle qualità differenziali di questo percorso di studi.

La padronanza della materia è completata dall'efficacia del disegno metodologico. Sviluppato da un team multidisciplinare di esperti di *e-learning*, integra gli ultimi progressi nella 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, mediante il quale lo specialista deve cercare di risolvere le diverse situazioni che gli si presentano durante il corso. Al fine di raggiungere questo obiettivo in modalità remota, useremo la pratica online: grazie all'aiuto di un innovativo sistema di video interattivi e il Learning from an Expert acquisirai conoscenze come se stessi vivendo la situazione che è oggetto di apprendimento. Un concetto che permetterà di integrare e fissare l'apprendimento in modo più realistico e permanente.

*Un Master Privato che ti permetterà di lavorare in tutti gli ambiti della Creazione di Interfacce e Applicazioni in Rete con la solvibilità di un professionista di alto livello.*

*Con l'esperienza di professionisti del settore che ti daranno una conoscenza reale, immediata e concreta di questo campo di lavoro.*



# 02 Obiettivi

L'obiettivo di TECH è fornire ai professionisti informazioni aggiornate sulla creazione di interfacce e applicazioni in rete. In questo modo, l'ingegnere di sistemi acquisirà nuove competenze per la creazione di programmi informatici più completi. Un obiettivo che, in pochi mesi, può essere raggiunto attraverso un programma di studi incentrato sulle attuali esigenze del settore.





“

*Amplia le tue conoscenze in informatica e ingegneria del software e preparati a competere con i migliori del settore"*



## Obiettivi generali

---

- ◆ Prepararsi 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

“

*Un percorso di studio e crescita professionale che ti proietterà verso una maggiore competitività nel mercato del lavoro”*





## Obiettivi specifici

---

### Modulo 1. Interazione uomo-macchina

- ◆ Acquisire solide conoscenze relative all'interazione uomo-macchina e alla creazione di interfacce utilizzabili
- ◆ Capire l'importanza dell'usabilità nelle applicazioni e perché bisogna tenerne conto quando si progetta il software
- ◆ Comprendere i vari tipi di diversità umana, le limitazioni che comportano e come adattare le interfacce in base alle esigenze specifiche di ciascuno di essi
- ◆ Imparare il processo di progettazione di un'interfaccia, dall'analisi dei requisiti alla valutazione, passando per le diverse fasi intermedie necessarie a creare un'interfaccia adeguata
- ◆ Conoscere le diverse linee guida sull'accessibilità, gli standard che le stabiliscono e gli strumenti che ci permettono di valutarle
- ◆ Comprendere i diversi metodi di interazione con il computer, utilizzando periferiche e dispositivi

### Modulo 2. Database

- ◆ Imparare le diverse applicazioni e finalità dei sistemi di database, nonché il loro funzionamento e la loro architettura
- ◆ Comprendere il modello relazionale, dalla sua struttura e operazioni all'algebra relazionale estesa
- ◆ Imparare in modo approfondito cosa sono i database SQL, come funzionano, la definizione dei dati e la creazione di query, dalle più elementari alle più avanzate
- ◆ Imparare a progettare database utilizzando il modello entità-relazionale, a creare diagrammi e a conoscere le caratteristiche del modello E-R esteso
- ◆ Approfondire la progettazione di database relazionali, analizzando le diverse forme normali e gli algoritmi di decomposizione
- ◆ Porre le basi per comprendere il funzionamento dei database NoSQL e introdurre il database Mongo DB

### Modulo 3. Sviluppo delle applicazioni in rete

- ◆ Imparare le caratteristiche del linguaggio di markup HTML e il suo utilizzo nella creazione di siti web insieme ai fogli di stile CSS
- ◆ Imparare a utilizzare il linguaggio di programmazione orientato al browser JavaScript e alcune delle sue caratteristiche principali
- ◆ Comprendere i concetti di programmazione orientata ai componenti e di architettura dei componenti
- ◆ Imparare a utilizzare il *Framework Front-End Bootstrap* per la progettazione di siti web
- ◆ Comprendere la struttura del modello controller-view nello sviluppo di siti web dinamici
- ◆ Conoscere l'architettura orientata ai servizi e le basi del protocollo HTTP

### Modulo 4. Software libero e conoscenza aperta

- ◆ Imparare i concetti di Software Libero e Conoscenza Aperta, nonché i diversi tipi di licenze associate
- ◆ Conoscere i principali strumenti gratuiti disponibili in diversi settori come i sistemi operativi, la gestione aziendale, i gestori di contenuti, la creazione di contenuti multimediali
- ◆ Comprendere l'importanza e i vantaggi del software libero nel mondo degli affari, sia in termini di caratteristiche che di costi
- ◆ Approfondire la conoscenza del sistema operativo GNU/Linux, delle diverse distribuzioni esistenti e di come realizzarne adattamenti personalizzati
- ◆ Conoscere il funzionamento e lo sviluppo di WordPress, dato che questo CMS rappresenta più del 35% dei siti web attivi nel mondo, e più del 60% nel caso particolare dei CMS
- ◆ Comprendere il funzionamento del sistema operativo dei dispositivi mobili Android e le basi dello sviluppo di applicazioni mobili sia in modo nativo che come con *Frameworks* multiplatforma

### Modulo 5. Database avanzati

- ◆ Introdurre i diversi sistemi di database attualmente disponibili sul mercato
- ◆ Apprendere l'uso di XML e dei database per il web
- ◆ Comprendere il funzionamento di database avanzati come i database paralleli e distribuiti
- ◆ Comprendere l'importanza dell'indicizzazione e dell'associazione nei sistemi di database
- ◆ Comprendere il funzionamento dei sistemi di elaborazione transazionale e i sistemi di recupero
- ◆ Acquisire conoscenze relative ai database non relazionali e al data mining

### Modulo 6. Ingegneria del software

- ◆ Comprendere il quadro dell'ingegneria del software e lo standard ISO/IEC 12207
- ◆ Imparare le caratteristiche del processo di sviluppo software unificato e della pianificazione nel contesto dello sviluppo software agile
- ◆ Conoscere i diversi stili di progettazione del software distribuito e architetture software orientate ai servizi
- ◆ Imparare i concetti essenziali della progettazione di interfacce grafiche
- ◆ Comprendere le basi dello sviluppo di applicazioni web
- ◆ Conoscere le strategie e le tecniche di test del software, i fattori di qualità del software e le diverse metriche utilizzate

### Modulo 7. Programmazione avanzata

- ◆ Approfondire la conoscenza della programmazione, soprattutto in relazione alla programmazione orientata agli oggetti, e dei diversi tipi di relazioni tra classi esistenti
- ◆ Conoscere i diversi modelli di progettazione per i problemi orientati agli oggetti
- ◆ Imparare la programmazione guidata dagli eventi e lo sviluppo di interfacce utente con Qt

- ◆ Acquisire le conoscenze essenziali della programmazione concorrente, dei processi e dei thread
- ◆ Imparare a gestire l'uso dei thread e della sincronizzazione, nonché a risolvere i problemi più comuni della programmazione concorrente
- ◆ Comprendere l'importanza della documentazione e dei test nello sviluppo del software

### **Modulo 8. Riutilizzo di software**

- ◆ Conoscere il panorama del riutilizzo di software
- ◆ Apprendere i diversi modelli relativi al riutilizzo del software, in termini di progettazione, creazione, struttura e comportamento
- ◆ Introdurre il concetto di *Framework*, nonché conoscerne le principali tipologie come quelle destinate alla progettazione di interfacce grafiche, allo sviluppo di applicazioni web e alla gestione della persistenza degli oggetti nei database
- ◆ Comprendere il funzionamento del pattern Model-View-Controller (MVC), ampiamente utilizzato

### **Modulo 9. 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 informata
- ◆ 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 10. Ingegneria dei Software Avanzata**

- ◆ Acquisire una conoscenza approfondita delle diverse metodologie agili utilizzate nell'ingegneria del software
- ◆ Imparare a sviluppare utilizzando Scrum, la programmazione estrema e le tecniche di sviluppo software basate sul riuso
- ◆ Comprendere i diversi modelli di architetture di sistema e di progettazione del software, nonché l'architettura delle applicazioni cloud
- ◆ Imparare a realizzare prove al software, con metodologie come *Test Driven Development*, *Acceptance Test Driven Development*, *Behavior Driven Development*, BDD e *Cucumber*
- ◆ Approfondire la comprensione del miglioramento del processo di sviluppo del software e della sua qualità utilizzando gli standard ISO/IEC
- ◆ Introdurre il concetto di DevOps e le sue pratiche principali

# 03

# Competenze

Questo programma sulla Creazione di Interfacce e Applicazioni in Rete è stato creato per offrire uno strumento di specializzazione avanzata per il professionista. Il suo piano di studi intensivo contribuirà in modo significativo allo sviluppo e al lavoro dei programmatori web e dei portali web, sfruttando le risorse internet esistenti e il software libero, con un orientamento pratico e utile.





“

*Con le competenze che questo Master Privato ti permetterà di acquisire, potrai iniziare a sviluppare applicazioni e lavorare come ingegnere del software, con la sicurezza di una preparazione completa e aggiornata"*



## Competenza generale

---

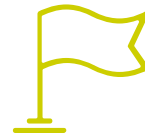
- ◆ Acquisire le competenze necessarie per la pratica professionale dell'ingegneria informatica con la conoscenza di tutti i fattori necessari per svolgerla con qualità e solvibilità

“

*Un'opportunità creata per gli esperti che cercano un programma intensivo ed efficace con cui compiere un significativo passo avanti nella propria professione”*







## Competenze specifiche

---

- ◆ Avere una conoscenza approfondita di tutte le sfaccettature dell'interazione uomo-macchina e di come queste coinvolgano gli sviluppi informatici
- ◆ Essere abili nell'uso delle banche dati
- ◆ Sviluppare diversi tipi di applicazioni di rete
- ◆ Descrivere e sfruttare il software libero e la conoscenza aperta sul web
- ◆ Lavorare come ingegnere del software
- ◆ Controllare l'uso di database avanzati
- ◆ Eseguire una programmazione avanzata
- ◆ Saper riutilizzare il software
- ◆ Creare interfacce e applicazioni di rete
- ◆ Padroneggiare i diversi sistemi di lavoro dell'ingegneria del software avanzata

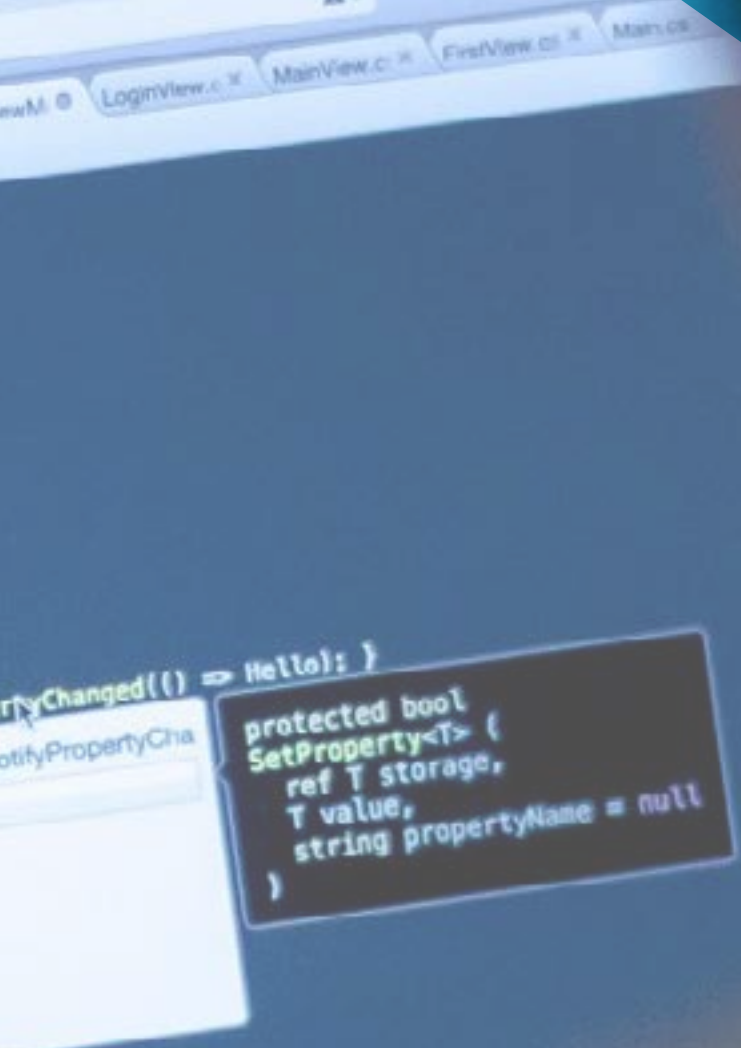
# 04

## Struttura e contenuti

I contenuti di questo Master Privato sono stati sviluppati dai diversi esperti del programma, con un obiettivo chiaro: garantire che gli studenti acquisiscano tutte le competenze necessarie per diventare veri esperti in questo campo.

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





“

*Un programma educativo molto completo, strutturato in unità didattiche esaustive e in base agli ultimi progressi del settore, orientate a un apprendimento compatibile con la tua vita personale e professionale"*

## Modulo 1. Interazione uomo-macchina

- 1.1. Introduzione all'interazione uomo-macchina
  - 1.1.1. Che cos'è l'interazione uomo-macchina
  - 1.1.2. Rapporto dell'interazione uomo-macchina con altre discipline
  - 1.1.3. L'interfaccia utente
  - 1.1.4. Usabilità e accessibilità
  - 1.1.5. Esperienza utente e design incentrato sull'utente
- 1.2. Il computer e l'interazione: interfaccia utente e paradigmi di interazione
  - 1.2.1. L'interazione
  - 1.2.2. Paradigmi di interazione e stili di interazione
  - 1.2.3. Evoluzione delle interfacce utente
  - 1.2.4. Interfacce utente classica: WIMP/GUI, comandi, voce, realtà virtuale
  - 1.2.5. Interfacce utente innovative: mobile, portatile, collaborativa, collaborativa, BCI
- 1.3. Il fattore umano: aspetti psicologici e cognitivi
  - 1.3.1. L'importanza del fattore umano nell'interazione
  - 1.3.2. Elaborazione dell'informazione umana
  - 1.3.3. L'ingresso e l'uscita delle informazioni: visive, uditive e tattili
  - 1.3.4. Percezione e attenzione
  - 1.3.5. Conoscenza e modelli mentali: rappresentazione, organizzazione e acquisizione
- 1.4. Il fattore umano: limitazioni sensoriali e fisiche
  - 1.4.1. Diversità funzionale, disabilità e menomazione
  - 1.4.2. Diversità visiva
  - 1.4.3. Diversità dell'udito
  - 1.4.4. Diversità cognitiva
  - 1.4.5. Diversità del motore
  - 1.4.6. Il caso degli immigrati digitali
- 1.5. Il processo di progettazione (I): analisi dei requisiti per la progettazione dell'interfaccia utente
  - 1.5.1. Design incentrato sull'utente
  - 1.5.2. Che cos'è l'analisi dei requisiti?
  - 1.5.3. Raccolta di informazioni
  - 1.5.4. Analisi e interpretazione delle informazioni
  - 1.5.5. Analisi di usabilità e accessibilità



- 1.6. Il processo di progettazione (II): prototipazione e analisi del compito
    - 1.6.1. Progetto concettuale
    - 1.6.2. Prototipazione
    - 1.6.3. Analisi gerarchica dei compiti
  - 1.7. Il processo di progettazione (III): la valutazione
    - 1.7.1. La valutazione nel processo di progettazione: obiettivi e metodi
    - 1.7.2. Metodi di valutazione senza utenti
    - 1.7.3. Metodi di valutazione con gli utenti
    - 1.7.4. Standard e norme di valutazione
  - 1.8. Accessibilità: definizione e linee guida
    - 1.8.1. Accessibilità e design universale
    - 1.8.2. L'iniziativa WAI e le linee guida WCAG
    - 1.8.3. Linee guida WCAG 2.0 e 2.1
  - 1.9. Accessibilità: valutazione e diversità funzionale
    - 1.9.1. Strumenti di valutazione dell'accessibilità del web
    - 1.9.2. Accessibilità e diversità funzionale
  - 1.10. Il computer e l'interazione: periferiche e dispositivi
    - 1.10.1. Dispositivi e periferiche tradizionali
    - 1.10.2. Dispositivi e periferiche alternative
    - 1.10.3. Telefoni cellulari e tablet
    - 1.10.4. Diversità funzionale, interazione e periferiche
- 
- Modulo 2. Database**
- 2.1. Applicazioni e scopi dei sistemi di database
    - 2.1.1. Applicazioni di diversi sistemi di database
    - 2.1.2. Scopo dei diversi sistemi di database
    - 2.1.3. Visione dei dati
  - 2.2. Database e architettura
    - 2.2.1. Database relazionale
    - 2.2.2. Progettazione di database
    - 2.2.3. Database a oggetti e semi-strutturati
    - 2.2.4. Memorizzazione dei dati e interrogazioni
    - 2.2.5. Gestione delle transazioni
    - 2.2.6. Estrazione e analisi dei dati
    - 2.2.7. Architettura del database
  - 2.3. Il modello relazionale: struttura, operazioni e algebra relazionale estesa
    - 2.3.1. La struttura dei DB relazionali
    - 2.3.2. Operazioni fondamentali dell'algebra relazionale
    - 2.3.3. Altre operazioni dell'algebra relazionale
    - 2.3.4. Operazioni di algebra relazionale estesa
    - 2.3.5. Valori nulli
    - 2.3.6. Modifica del database
  - 2.4. SQL (I)
    - 2.4.1. Cos'è SQL?
    - 2.4.2. La definizione di dati
    - 2.4.3. La struttura di base delle query SQL
    - 2.4.4. Operazioni sugli insiemi
    - 2.4.5. Funzioni di aggregazione
    - 2.4.6. Valori nulli
  - 2.5. SQL (II)
    - 2.5.1. Subquery annidate
    - 2.5.2. Query complesse
    - 2.5.3. Viste
    - 2.5.4. Cursori
    - 2.5.5. Query complesse
    - 2.5.6. Trigger
  - 2.6. Progettazione di database e modello E-R
    - 2.6.1. Panoramica del processo di progettazione
    - 2.6.2. Modello entità-relazione
    - 2.6.3. Restrizioni
  - 2.7. Diagrammi entità-relazione
    - 2.7.1. Diagrammi entità-relazione
    - 2.7.2. Aspetti di progettazione delle entità e delle relazioni
    - 2.7.3. Insiemi di entità deboli
  - 2.8. Il modello esteso di entità-relazione
    - 2.8.1. Caratteristiche del modello E-R esteso
    - 2.8.2. Progettazione del database
    - 2.8.3. Riduzione a schemi relazionali

- 2.9. Progettazione di database relazionali
  - 2.9.1. Caratteristiche di un buon progetto relazionale
  - 2.9.2. Domini atomici e prima forma normale (1FN)
  - 2.9.3. Decomposizione mediante dipendenze funzionali
  - 2.9.4. Teoria della dipendenza funzionale
  - 2.9.5. Algoritmi di decomposizione
  - 2.9.6. Decomposizione con dipendenze multivariate
  - 2.9.7. Altre forme normali
  - 2.9.8. Processo di progettazione del database
- 2.10. Database NoSQL
  - 2.10.1. Cosa sono i database NoSQL?
  - 2.10.2. Analisi delle diverse opzioni NoSQL e delle loro caratteristiche
  - 2.10.3. Mongo DB

### Modulo 3. Sviluppo delle applicazioni in rete

- 3.1. Linguaggi di mercato HTML5
  - 3.1.1. Nozioni di base sulla HTML
  - 3.1.2. Nuovi elementi HTML 5
  - 3.1.3. Moduli: nuovi controlli
- 3.2. Introduzione ai fogli di stile CSS
  - 3.2.1. Primi passi con CSS
  - 3.2.2. Introduzione ai CSS3
- 3.3. Linguaggio di scripting del browser: JavaScript
  - 3.3.1. Nozioni di base di JavaScript
  - 3.3.2. DOM
  - 3.3.3. Eventi
  - 3.3.4. JQuery
  - 3.3.5. Ajax
- 3.4. Concetto di programmazione orientata ai componenti
  - 3.4.1. Contesto
  - 3.4.2. Componenti e interfacce
  - 3.4.3. Stati di un componente

- 3.5. Architettura dei componenti
  - 3.5.1. Architetture attuali
  - 3.5.2. Integrazione e distribuzione dei componenti
- 3.6. *Framework Front-End: Bootstrap*
  - 3.6.1. Design con la rete
  - 3.6.2. Moduli
  - 3.6.3. Componenti
- 3.7. Controllore della vista del modello
  - 3.7.1. Metodi di sviluppo web
  - 3.7.2. Pattern di progettazione: MVC
- 3.8. Tecnologie *Grid* informative
  - 3.8.1. Aumento delle risorse informatiche
  - 3.8.2. Concetto di tecnologia *Grid*
- 3.9. Architetture orientate ai servizi
  - 3.9.1. SOA e servizi web
  - 3.9.2. Topologia del servizio web
  - 3.9.3. Piattaforme di servizi web
- 3.10. Protocollo HTTP
  - 3.10.1. Messaggi
  - 3.10.2. Sessioni persistenti
  - 3.10.3. Sistema crittografico
  - 3.10.4. Funzionamento del protocollo HTTPS

### Modulo 4. Software libero e conoscenza aperta

- 4.1. Introduzione al software libero
  - 4.1.1. Storia del software libero
  - 4.1.2. "Libertà" del software
  - 4.1.3. Licenze per l'uso di strumenti software
  - 4.1.4. Proprietà intellettuale del software
  - 4.1.5. Qual è la motivazione per l'utilizzo del software libero?
  - 4.1.6. Miti del software libero
  - 4.1.7. Top500

- 4.2. Conoscenza aperta e licenze CC
  - 4.2.1. Concetti di base
  - 4.2.2. Licenze *Creative Commons*
  - 4.2.3. Altre licenze di contenuto
  - 4.2.4. Wikipedia e altri progetti di conoscenza aperta
- 4.3. Principali strumenti di software libero
  - 4.3.1. Sistemi operativi
  - 4.3.2. Applicazioni per ufficio
  - 4.3.3. Applicazioni di gestione aziendale
  - 4.3.4. Gestori di Contenuti Web
  - 4.3.5. Strumenti per la creazione di contenuti multimediali
  - 4.3.6. Altre applicazioni
- 4.4. L'azienda: il software libero e i suoi costi
  - 4.4.1. Software libero: Sì o no?
  - 4.4.2. Verità e bugie sul software libero
  - 4.4.3. Software aziendale basato su software libero
  - 4.4.4. Software aziendale basato su software libero
  - 4.4.5. Modelli di software liberi
- 4.5. Il sistema operativo GNU/Linux
  - 4.5.1. Architettura
  - 4.5.2. Struttura della directory base
  - 4.5.3. Interfaccia e implementazione del file system
  - 4.5.4. Rappresentazione interna dei file
- 4.6. Il sistema operativo mobile Android
  - 4.6.1. Storia
  - 4.6.2. Architettura
  - 4.6.3. Forks di Android
  - 4.6.4. Introduzione allo sviluppo di Android
  - 4.6.5. *Frameworks* per lo sviluppo di applicazioni mobili
- 4.7. Creazione di siti web con WordPress
  - 4.7.1. Caratteristiche e struttura di WordPress
  - 4.7.2. Creazione di siti su wordpress.com
  - 4.7.3. Installazione e configurazione di WordPress in un server proprio
  - 4.7.4. Installazione di Plugins ed estensione di WordPress
  - 4.7.5. Creazione di Plugins per WordPress
  - 4.7.6. Creazione di temi per WordPress
- 4.8. Tendenze del software libero
  - 4.8.1. Ambienti nel cloud
  - 4.8.2. Strumenti di monitoraggio
  - 4.8.3. Sistemi operativi
  - 4.8.4. Big Data e Open Data 2.0
  - 4.8.5. Informatica quantistica
- 4.9. Controllo delle versioni
  - 4.9.1. Concetti di base
  - 4.9.2. Git
  - 4.9.3. Servizi Git in cloud e self-hosted
  - 4.9.4. Altri sistemi di controllo della versione
- 4.10. Distribuzione di GNU/Linux personalizzati
  - 4.10.1. Principali distribuzioni
  - 4.10.2. Distribuzioni derivate da Debian
  - 4.10.3. Creazione pacchetti deb
  - 4.10.4. Modifica della distribuzione
  - 4.10.5. Creazione di immagini ISO

## Modulo 5. Database avanzati

- 5.1. Scopo dei diversi sistemi di database
  - 5.1.1. Rassegna storica
  - 5.1.2. Database gerarchici
  - 5.1.3. Database di rete
  - 5.1.4. Database relazionali
  - 5.1.5. Database non relazionali

- 5.2. XML e database per il web
  - 5.2.1. Convalida dei documenti XML
  - 5.2.2. Trasformazioni dei documenti XML
  - 5.2.3. Memorizzazione di dati XML
  - 5.2.4. Database relazionali XML
  - 5.2.5. SQL/XML
  - 5.2.6. Database XML nativi
- 5.3. Basi di dati parallele
  - 5.3.1. Sistemi paralleli
  - 5.3.2. Architetture di database paralleli
  - 5.3.3. Parallelismo delle query
  - 5.3.4. Parallelismo tra query
  - 5.3.5. Progettazione di sistemi paralleli
  - 5.3.6. Elaborazione parallela in SQL
- 5.4. Database distribuiti
  - 5.4.1. Sistemi distribuiti
  - 5.4.2. Archiviazione distribuita
  - 5.4.3. Disponibilità
  - 5.4.4. Elaborazione distribuita delle query
  - 5.4.5. Fornitori di database distribuiti
- 5.5. Indicizzazione e associazione
  - 5.5.1. Indici ordinati
  - 5.5.2. Indici densi e radi
  - 5.5.3. Indici multilivello
  - 5.5.4. Aggiornamento dell'indice
  - 5.5.5. Associazione statica
  - 5.5.6. Come utilizzare gli indici nei database
- 5.6. Introduzione all'elaborazione transazionale
  - 5.6.1. Stati di una transazione
  - 5.6.2. Implementazione dell'atomicità e della durata
  - 5.6.3. Sequenzialità
  - 5.6.4. Recuperabilità
  - 5.6.5. Implementazione dell'isolamento





- 
- 5.7. Sistemi di recupero
    - 5.7.1. Classificazione dei guasti
    - 5.7.2. Strutture di archiviazione
    - 5.7.3. Recupero e atomicità
    - 5.7.4. Recupero basato sul record storico
    - 5.7.5. Transazioni e recupero concorrenti
    - 5.7.6. Alta disponibilità nei database
  - 5.8. Esecuzione ed elaborazione di query
    - 5.8.1. Costo di una query
    - 5.8.2. Operazione di selezione
    - 5.8.3. Ordinamento
    - 5.8.4. Introduzione all'ottimizzazione delle query
    - 5.8.5. Monitoraggio delle prestazioni
  - 5.9. Database non relazionali
    - 5.9.1. Database orientati ai documenti
    - 5.9.2. Database orientati alla rete
    - 5.9.3. Database a valore-chiave
  - 5.10. Data Warehouse, OLAP e data mining
    - 5.10.1. Componenti di un data warehouse
    - 5.10.2. Architettura di un data warehouse
    - 5.10.3. OLAP
    - 5.10.4. Funzionalità di data mining
    - 5.10.5. Altri tipi di estrazione mineraria

## Modulo 6. Ingegneria del software

- 6.1. Struttura dell'ingegneria del software
  - 6.1.1. Caratteristiche del software
  - 6.1.2. Principali processi di ingegneria del software
  - 6.1.3. Modelli di processo di sviluppo software
  - 6.1.4. Quadro di riferimento standard per il processo di sviluppo del software: lo standard ISO/IEC 12207

- 6.2. Processo unificato di sviluppo software
  - 6.2.1. Processo unificato
  - 6.2.2. Dimensioni del processo unificato
  - 6.2.3. Processo di sviluppo guidato dai casi d'uso
  - 6.2.4. Flussi di lavoro unificati fondamentali per i processi
- 6.3. Pianificazione nel contesto dello sviluppo agile del software
  - 6.3.1. Caratteristiche dello sviluppo agile del software
  - 6.3.2. Diversi orizzonti temporali di pianificazione nello sviluppo agile
  - 6.3.3. Quadro di sviluppo Agile Scrum e pianificazione degli orizzonti temporali
  - 6.3.4. Storie dell'utente come unità di pianificazione e stima
  - 6.3.5. Tecniche comuni per ricavare una stima
  - 6.3.6. Scale di interpretazione delle stime
  - 6.3.7. *Planning Poker*
  - 6.3.8. Tipi di pianificazione comuni: pianificazione della consegna e dell'iterazione
- 6.4. Stili di progettazione del software distribuito e architetture software orientate ai servizi
  - 6.4.1. Modelli di comunicazione nei sistemi software distribuiti
  - 6.4.2. Livello intermedio o middleware
  - 6.4.3. Modelli di architettura per sistemi distribuiti
  - 6.4.4. Processo generale di progettazione dei servizi software
  - 6.4.5. Aspetti di progettazione dei servizi software
  - 6.4.6. Composizione dei servizi
  - 6.4.7. Architettura dei servizi web
  - 6.4.8. Componenti infrastrutturali e SOA
- 6.5. Introduzione allo sviluppo di software guidato da modelli
  - 6.5.1. Il concetto di modello
  - 6.5.2. Sviluppo di software guidato da modelli
  - 6.5.3. Quadro di sviluppo guidato dal modello MDA
  - 6.5.4. Elementi di un modello di trasformazione
- 6.6. Progettazione dell'interfaccia grafica
  - 6.6.1. Principi di progettazione dell'interfaccia utente
  - 6.6.2. Modelli di progettazione architettonica per sistemi interattivi: Model-View-Controller (MVC)
  - 6.6.3. Esperienza dell'utente (UX *User Experience*)
  - 6.6.4. Design incentrato sull'utente
  - 6.6.5. Processo di analisi e progettazione dell'interfaccia utente
  - 6.6.6. Usabilità delle interfacce utente
  - 6.6.7. Accessibilità delle interfacce utente
- 6.7. Disegno delle applicazioni web
  - 6.7.1. Caratteristiche delle applicazioni web
  - 6.7.2. Interfaccia utente di un'applicazione web
  - 6.7.3. Design della navigazione
  - 6.7.4. Protocollo di interazione di base per le applicazioni web
  - 6.7.5. Stili di architettura per applicazioni web
- 6.8. Strategie e tecniche di test del software e fattori di qualità dello stesso
  - 6.8.1. Strategie di prova
  - 6.8.2. Progettazione dei casi di prova
  - 6.8.3. Relazione costo-qualità
  - 6.8.4. Modelli di qualità
  - 6.8.5. Famiglia di norme ISO/IEC 25000 (SQuaRE)
  - 6.8.6. Modello di qualità del prodotto (ISO 2501n)
  - 6.8.7. Modello di qualità dei dati (ISO 2501n)
  - 6.8.8. Gestione della qualità del software
- 6.9. Introduzione alle metriche nell'ingegneria del software
  - 6.9.1. Concetti di base: misure, metriche e indicatori
  - 6.9.2. Tipi di metriche nell'ingegneria del software
  - 6.9.3. Processo di misurazione
  - 6.9.4. ISO 25024. Metriche esterne e di qualità in uso
  - 6.9.5. Metrica orientata agli oggetti

- 6.10. Manutenzione e reingegnerizzazione del software
  - 6.10.1. Processo di manutenzione
  - 6.10.2. Quadro standard del processo di manutenzione: ISO/EIEC 14764
  - 6.10.3. Modello del processo di reingegnerizzazione del software
  - 6.10.4. Ingegneria inversa

## Modulo 7. Programmazione avanzata

- 7.1. Introduzione alla programmazione orientata agli oggetti
  - 7.1.1. Introduzione alla programmazione orientata agli oggetti
  - 7.1.2. Progettazione delle lezioni
  - 7.1.3. Introduzione a UML per la modellazione dei problemi
- 7.2. Relazioni tra lezioni
  - 7.2.1. Astrazione ed ereditarietà
  - 7.2.2. Concetti avanzati di ereditarietà
  - 7.2.3. Polimorfismo
  - 7.2.4. Composizione e aggregazione
- 7.3. Introduzione ai design pattern per i problemi orientati agli oggetti
  - 7.3.1. Cosa sono i design pattern?
  - 7.3.2. Pattern *Factory*
  - 7.3.3. Pattern *Singleton*
  - 7.3.4. Pattern *Observer*
  - 7.3.5. Pattern *Composite*
- 7.4. Eccezioni
  - 7.4.1. Quali sono le eccezioni?
  - 7.4.2. Gestione e acquisizione delle eccezioni
  - 7.4.3. Avvio delle eccezioni
  - 7.4.4. Creazione di eccezioni
- 7.5. Interfacce utente
  - 7.5.1. Introduzione a Qt
  - 7.5.2. Posizionamento
  - 7.5.3. Cosa sono gli eventi?
  - 7.5.4. Eventi: definizione e acquisizione
  - 7.5.5. Sviluppo di interfacce utente
- 7.6. Introduzione alla programmazione concorrente
  - 7.6.1. Introduzione alla programmazione concorrente
  - 7.6.2. Il concetto di processo e di thread
  - 7.6.3. Interazione tra processi o thread
  - 7.6.4. Thread in C++
  - 7.6.5. Vantaggi e svantaggi della programmazione concorrente
- 7.7. Gestione e sincronizzazione dei thread
  - 7.7.1. Ciclo di vita di un thread
  - 7.7.2. La classe *Thread*
  - 7.7.3. Pianificazione del thread
  - 7.7.4. Gruppi di thread
  - 7.7.5. Thread di tipo demoniaco
  - 7.7.6. Sincronizzazione
  - 7.7.7. Meccanismi di bloccaggio
  - 7.7.8. Meccanismi di comunicazione
  - 7.7.9. Monitor
- 7.8. Problemi comuni nella programmazione concorrente
  - 7.8.1. Il problema dei produttori-consumatori
  - 7.8.2. Il problema dei lettori e degli scrittori
  - 7.8.3. Il problema dei filosofi a cena
- 7.9. Documentazione e test del software
  - 7.9.1. Perché è importante documentare il software?
  - 7.9.2. Documentazione di progettazione
  - 7.9.3. Utilizzo di strumenti per la documentazione
- 7.10. Test di software
  - 7.10.1. Introduzione al test del software
  - 7.10.2. Tipi di test
  - 7.10.3. Test dell'unità
  - 7.10.4. Test di integrità
  - 7.10.5. Test di convalida
  - 7.10.6. Test del sistema

## Modulo 8. Riutilizzo di software

- 8.1. Panoramica del riutilizzo di software
  - 8.1.1. In che consiste il riutilizzo di software
  - 8.1.2. Vantaggi e svantaggi del riutilizzo di software
  - 8.1.3. Principali tecniche di riutilizzo di software
- 8.2. Introduzione ai modelli di progettazione
  - 8.2.1. Che cos'è un modello di progettazione?
  - 8.2.2. Catalogo dei principali modelli di progettazione
  - 8.2.3. Come utilizzare i modelli per risolvere i problemi di progettazione
  - 8.2.4. Come selezionare il miglior modello di progettazione
- 8.3. Modelli di creazione
  - 8.3.1. Modelli di creazione
  - 8.3.2. Modello *Abstract Factory*
  - 8.3.3. Esempio di implementazione del modello *Abstract Factory*
  - 8.3.4. Modello Builder
  - 8.3.5. Esempio di implementazione del metodo Builder
  - 8.3.6. Modello *Abstract Factory* vs Builder
- 8.4. Modelli di creazione (II)
  - 8.4.1. Modello *Factory Method*
  - 8.4.2. *Factory Method* vs. *Abstract Factory*
  - 8.4.3. Pattern Singleton
- 8.5. Modelli strutturali
  - 8.5.1. Modelli strutturali
  - 8.5.2. Modello *Adapter*
  - 8.5.3. Modello *Bridge*
- 8.6. Modelli strutturali (II)
  - 8.6.1. Pattern *Composite*
  - 8.6.2. Modello Decorator
- 8.7. Modelli strutturali (III)
  - 8.7.1. Modello Facade
  - 8.7.2. Modello Proxy

- 8.8. Modelli di comportamento
  - 8.8.1. Concetto di modelli di comportamento
  - 8.8.2. Modelli di comportamento: catena di responsabilità
  - 8.8.3. Modello di comportamento Ordine
- 8.9. Modelli di comportamento (II)
  - 8.9.1. Modello Interprete o *Interpreter*
  - 8.9.2. Modello di iterazione
  - 8.9.3. Modello di osservazione
  - 8.9.4. Modello di strategia
- 8.10. *Frameworks*
  - 8.10.1. Concetto di *Frameworks*
  - 8.10.2. Sviluppo mediante *Frameworks*
  - 8.10.3. Modello *Model View Controller*
  - 8.10.4. *Framework* per la progettazione di interfacce grafiche
  - 8.10.5. *Frameworks* per lo sviluppo di applicazioni web
  - 8.10.6. *Frameworks* per la gestione della persistenza degli oggetti nei database

## Modulo 9. Intelligenza Artificiale e Ingegneria della Conoscenza

- 9.1. Introduzione all'intelligenza artificiale e all'ingegneria della conoscenza
  - 9.1.1. Breve storia dell'intelligenza artificiale
  - 9.1.2. L'intelligenza Artificiale oggi
  - 9.1.3. Ingegneria della conoscenza
- 9.2. Ricerca
  - 9.2.1. Concetti di ricerca comuni
  - 9.2.2. Ricerca non informata
  - 9.2.3. Ricerca informata
- 9.3. Soddisfacibilità booleana, soddisfacibilità dei vincoli e pianificazione automatica
  - 9.3.1. Soddisfacibilità booleana
  - 9.3.2. Problemi di soddisfazione dei vincoli
  - 9.3.3. Pianificazione automatica e PDDL
  - 9.3.4. Pianificazione come ricerca euristica
  - 9.3.5. Pianificazione con il SAT

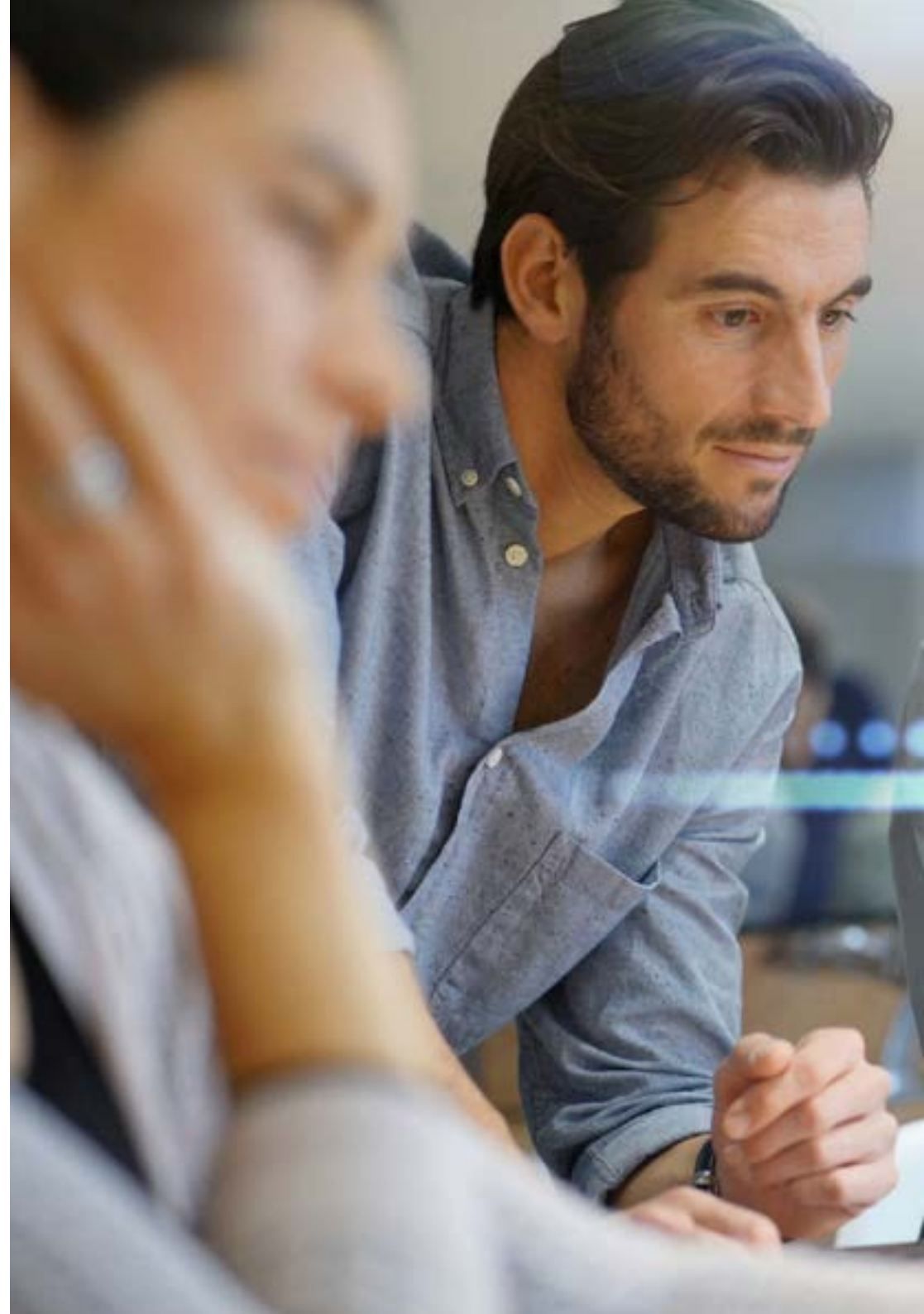
- 9.4. Intelligenza artificiale nei giochi
  - 9.4.1. Teoria dei giochi
  - 9.4.2. Potenziamento Minimax e Alfa-Beta
  - 9.4.3. Simulazione: Monte Carlo
- 9.5. Apprendimento supervisionato e non
  - 9.5.1. Introduzione all'apprendimento automatico
  - 9.5.2. Classificazione
  - 9.5.3. Regressione
  - 9.5.4. Convalida dei risultati
  - 9.5.5. Raggruppamento (*Clustering*)
- 9.6. Reti neurali
  - 9.6.1. Basi biologiche
  - 9.6.2. Modello computazionale
  - 9.6.3. Reti neurali supervisionate e non
  - 9.6.4. Percettrone semplice
  - 9.6.5. Percettrone multistrato
- 9.7. Algoritmi genetici
  - 9.7.1. Storia
  - 9.7.2. Base biologica
  - 9.7.3. Codifica dei problemi
  - 9.7.4. Generazione della popolazione iniziale
  - 9.7.5. Algoritmo principale e operatori genetici
  - 9.7.6. Valutazione degli individui: fitness
- 9.8. Thesauri, vocabolari, tassonomie
  - 9.8.1. Vocabolari
  - 9.8.2. Tassonomie
  - 9.8.3. Thesauri
  - 9.8.4. Ontologie
- 9.9. Rappresentazione della conoscenza: web semantico
  - 9.9.1. Web semantico
  - 9.9.2. Specifiche: RDF, RDFS e OWL
  - 9.9.3. Inferenza/ragionamento
  - 9.9.4. *Linked Data*

- 9.10. Sistemi esperti e DSS
  - 9.10.1. Sistemi esperti
  - 9.10.2. Sistemi di supporto decisionale

## Modulo 10. Ingegneria dei Software Avanzata

- 10.1. Introduzione alle Metodologie Agili
  - 10.1.1. Modelli di processo e metodologie
  - 10.1.2. Agilità e processi agili
  - 10.1.3. Manifesto Agile
  - 10.1.4. Alcune metodologie agili
  - 10.1.5. Agile vs Tradizionale
- 10.2. Scrum
  - 10.2.1. Origini e filosofia di Scrum
  - 10.2.2. Valori di Scrum
  - 10.2.3. Flusso del processo Scrum
  - 10.2.4. Ruoli di Scrum
  - 10.2.5. Artefatti di Scrum
  - 10.2.6. Eventi di Scrum
  - 10.2.7. Storie degli utenti
  - 10.2.8. Estensioni Scrum
  - 10.2.9. Stime agili
  - 10.2.10. Scalabilità di Scrum
- 10.3. Programmazione estrema
  - 10.3.1. Motivazione e panoramica di XP
  - 10.3.2. Il ciclo di vita in XP
  - 10.3.3. I cinque valori fondamentali
  - 10.3.4. Le dodici pratiche di base di XP
  - 10.3.5. Ruoli dei partecipanti
  - 10.3.6. XP industriale
  - 10.3.7. Valutazione critica di XP

- 10.4. Sviluppo del software basato sul riutilizzo
  - 10.4.1. Riutilizzo del software
  - 10.4.2. Livelli di riutilizzo del codice
  - 10.4.3. Tecniche di riutilizzo specifiche
  - 10.4.4. Sviluppo basato su componenti
  - 10.4.5. Benefici e problemi del riutilizzo
  - 10.4.6. Pianificazione del riutilizzo
- 10.5. Architettura di sistema e modelli di progettazione del software
  - 10.5.1. Design architettonico
  - 10.5.2. Modelli architettonici generali
  - 10.5.3. Architetture con tolleranza ai guasti
  - 10.5.4. Architetture di sistemi distribuiti
  - 10.5.5. Modelli di progettazione
  - 10.5.6. Modelli gamma
  - 10.5.7. Modelli di progettazione dell'interazione
- 10.6. Architettura di applicazioni nel cloud
  - 10.6.1. Fondamenti di *Cloud Computing*
  - 10.6.2. Qualità delle applicazioni nel cloud
  - 10.6.3. Stili di architettura
  - 10.6.4. Modelli di design
- 10.7. Test del software: TDD, ATDD e BDD
  - 10.7.1. Verifica e convalida del software
  - 10.7.2. Test del software
  - 10.7.3. *Test Driven Development* (TDD)
  - 10.7.4. *Acceptance Test Driven Development* (ATDD)
  - 10.7.5. *Behavior Driven Development* (BDD)
  - 10.7.6. BDD e *Cucumber*
- 10.8. Miglioramento dei processi software
  - 10.8.1. Miglioramento dei processi software
  - 10.8.2. Il miglioramento dei processi
  - 10.8.3. Modelli di maturità
  - 10.8.4. Il modello CMMI
  - 10.8.5. CMMI V2.0
  - 10.8.6. CMMI e Agile





- 10.9. La qualità del prodotto software: SQuaRE
  - 10.9.1. La qualità del software
  - 10.9.2. Modello di qualità del prodotto software
  - 10.9.3. Famiglia ISO/IEC 25000
  - 10.9.4. ISO/IEC 25010: modello e caratteristiche di qualità
  - 10.9.5. ISO/IEC 25012: la qualità dei dati
  - 10.9.6. ISO/IEC 25020: misurazione della qualità del software
  - 10.9.7. ISO/IEC 25022, 25023 e 25024: metriche di qualità del software e dei dati
  - 10.9.8. ISO/IEC 25040: valutazione del software
  - 10.9.9. Processo di certificazione
- 10.10. Introduzione a DevOps
  - 10.10.1. Concetto di DevOps
  - 10.10.2. Pratiche principali

“

*Un'esperienza unica, chiave  
e decisiva per incrementare  
il tuo sviluppo professionale”*

# 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.



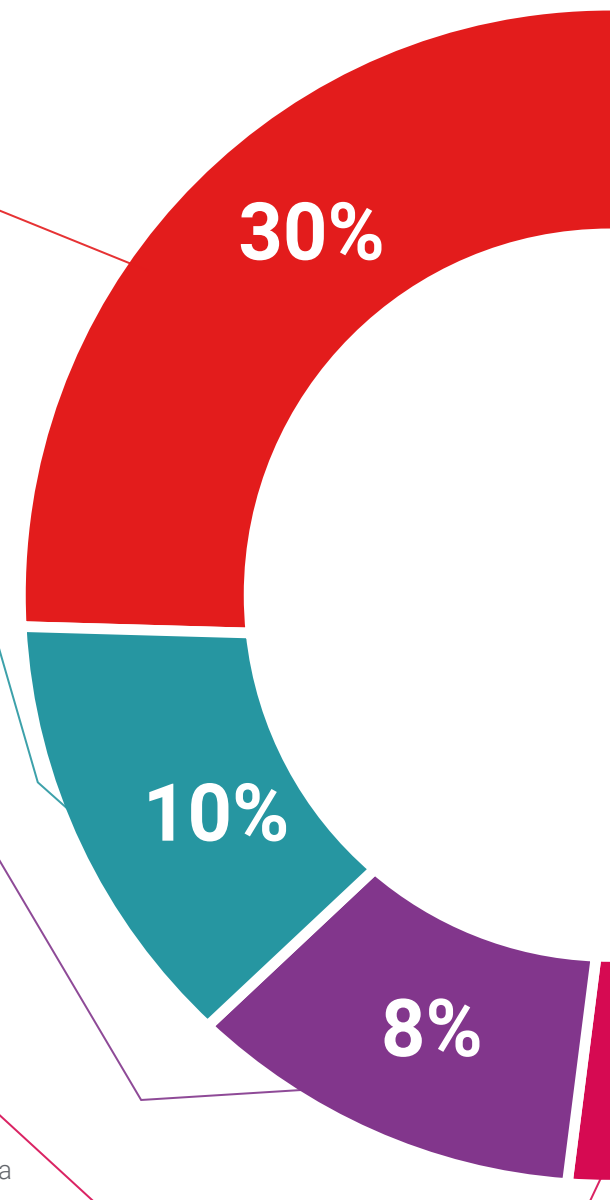
#### Pratiche di competenze e competenze

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 Creazione di Interfacce e Applicazioni in Rete garantisce, oltre alla preparazione più rigorosa e aggiornata, il conseguimento di una qualifica di Master Privato rilasciata da TECH Università Tecnologica.





“

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

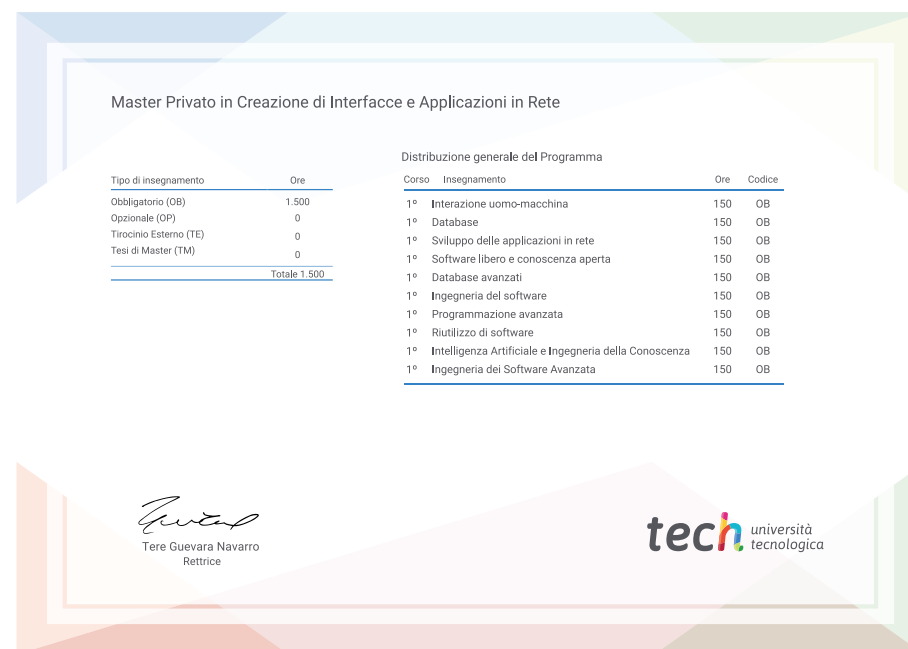
Questo **Master Privato in Creazione di Interfacce e Applicazioni in Rete** possiede il programma più completo e aggiornato del mercato.

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 Creazione di Interfacce e Applicazioni in Rete**

N. Ore Ufficiali: **1.500 o.**



\*Se lo studente dovesse richiedere che il suo diploma cartaceo sia provvisto di Apostille dell'Aia, TECH EDUCATION effettuerà 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 innovazione  
conoscenza presente qualità  
formazione online  
sviluppo istituzioni  
classe virtuale lingue

**tech** università  
tecnologica

**Master Privato**  
Creazione di Interfacce  
e Applicazioni in Rete

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

# Master Privato

Creazione di Interfacce  
e Applicazioni in Rete