Master Semipresenziale
Sviluppo di Software

tech università tecnologica

tech università tecnologica

Master Semipresenziale
Sviluppo di Software

Modalità: **Semipresenziale (Online + Tirocinio)**
Durata: **12 mesi**
Titolo: **TECH Università Tecnologica**
Accesso al sito web: **www.techtitute.com/it/informatica/master-semipresenziale/master-semipresenziale-sviluppo-software**

Indice

01

Presentazione

Lo Sviluppo di Software è diventato una componente cruciale dell'infrastruttura tecnologica nella maggior parte dei settori, dalla sanità alla finanza e all'intrattenimento. Con la crescente complessità delle applicazioni, gli informatici devono adattarsi a un ambiente in costante cambiamento. Ciò richiede che gli sviluppatori aggiornino frequentemente le loro conoscenze per rimanere al passo con i più recenti sviluppi in materie come linguaggi di programmazione o strategie per la creazione di algoritmi. In questo contesto, TECH presenta un innovativo corso universitario che approfondirà le più recenti innovazioni nel campo dello Sviluppo di Software.

Grazie a questo Master Semipresenziale, applicherai modelli di progettazione per risolvere una vasta gamma di problemi informatici e costruire soluzioni scalabili"

Lo Sviluppo di Software è diventato un fattore essenziale nell'economia digitale di oggi, con una crescente domanda di professionisti qualificati in tutto il mondo. Secondo l'Organizzazione per la Cooperazione e lo Sviluppo Economico, questo settore professionale dovrebbe registrare una crescita del 30% nel prossimo anno. Questo sottolinea l'importanza per i professionisti di informatica di rimanere aggiornati sulle ultime tendenze in questo campo. Altrimenti, gli sviluppatori potrebbero trovarsi di fronte a problemi come l'uso di metodologie obsolete che portano a processi meno efficienti.

In questo contesto, TECH propone un rivoluzionario Master Semipresenziale in Sviluppo di Software. Si tratta di un programma universitario che unisce in 1.920 ore il miglior contenuto teorico con 3 settimane di un tirocinio educativo pratico in un'entità di riferimento in questo settore. Il percorso accademico approfondirà aspetti come la Programmazione in C++, la creazione di database avanzati o la progettazione architettonica delle applicazioni. Tutto questo attraverso materiali didattici preparati da un team di insegnanti esperti, che includono una vasta gamma di risorse multimediali (come riassunti interattivi, casi di studio o video esplicativi) per garantire un aggiornamento piacevole. In questo modo, gli informatici potranno godere di un apprendimento totalmente progressivo e naturale, senza dover ricorrere a tecniche tradizionali come la memorizzazione.

Inoltre, il titolo universitario prevede che gli studenti svolgano una formazione pratica presso un'istituzione prestigiosa. Gli informatici parteciperanno attivamente ai progetti che si stanno sviluppando in quel momento. Si noti che un tutor specializzato guiderà gli studenti durante questo soggiorno presenziale, assicurando loro la realizzazione di un piano di attività che permetterà loro di ottimizzare le proprie competenze in base alle esigenze del mercato del lavoro attuale.

Questo **Master Semipresenziale in Sviluppo di Software** possiede il programma più completo e aggiornato del mercato. Le caratteristiche principali del programma sono:

- Sviluppo di oltre 100 casi di studio presentati da professionisti in Sviluppo di Software
- Contenuti grafici, schematici ed eminentemente pratici che forniscono informazioni scientifiche e pratiche sulle discipline essenziali per l'esercizio della professione
- Notizie sugli ultimi sviluppi in Sviluppo di Software
- Esercizi pratici che offrono un processo di autovalutazione per migliorare l'apprendimento.
- Lezioni teoriche, domande all'esperto, forum di discussione su argomenti controversi e lavori di riflessione individuale
- Contenuti disponibili da qualsiasi dispositivo fisso o mobile dotato di connessione a internet
- Possibilità di svolgere un tirocinio presso una delle migliori aziende del settore

Implementerai processi di assicurazione della qualità nella tua pratica per garantire l'affidabilità e le prestazioni del Software"

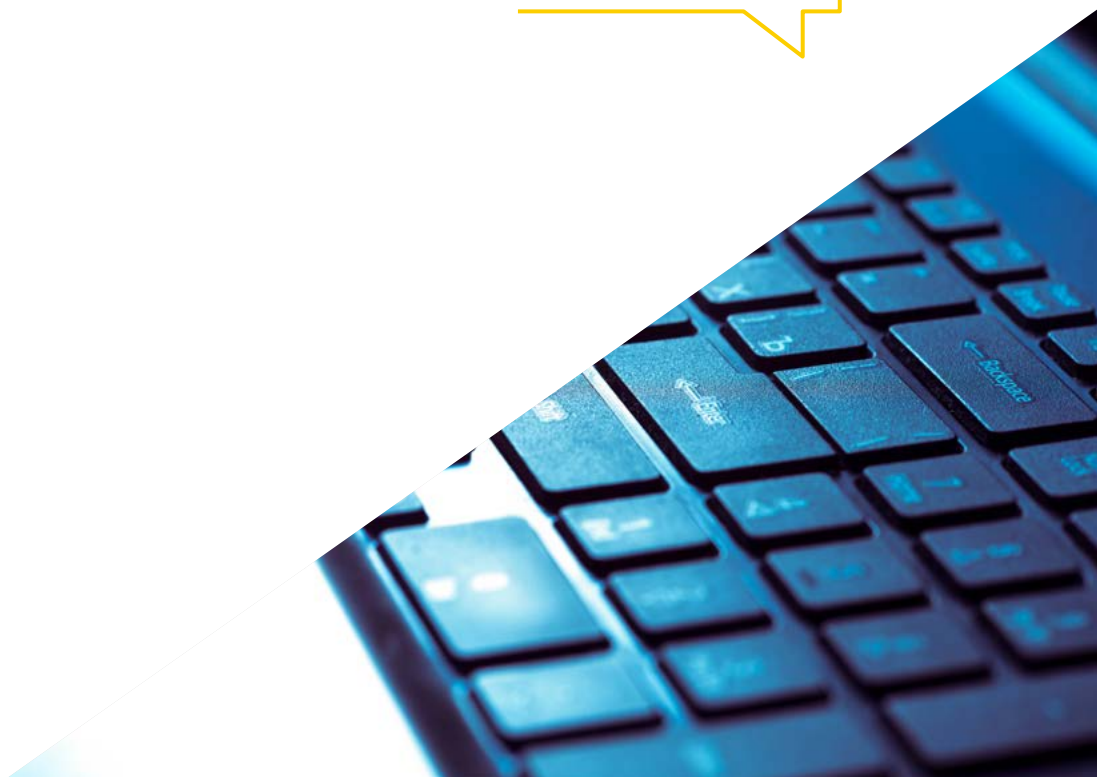
Frequenta un tirocinio educativo intensivo di 3 settimane in una società prestigiosa e acquisisce tutte le conoscenze per sperimentare un notevole salto di qualità professionale"

Questo titolo universitario includerà casi reali per avvicinare al massimo lo sviluppo del programma alla realtà della prassi informatica.

Avrai il supporto della più grande istituzione accademica online del mondo, TECH, con la più recente tecnologia educativa a tua disposizione.

In questa proposta di Master, di carattere professionalizzante e modalità semipresenziale, il programma è diretto all'aggiornamento dei professionisti dell'informatica che vogliono incorporare nella loro pratica le tecniche più avanzate per lo Sviluppo di Software. I contenuti sono basati sulle ultime prove scientifiche e orientati in modo didattico per integrare il sapere teorico nella pratica informatica e gli elementi teorico-pratici faciliteranno l'aggiornamento delle conoscenze.

Grazie ai contenuti multimediali elaborati con la più recente tecnologia educativa, permetteranno al professionista dell'informatica un apprendimento localizzato e contestuale, cioè un ambiente simulato che fornirà un apprendimento immersivo programmato per allenarsi in situazioni reali. La creazione di questo programma si basa sull'Apprendimento Basato su Problemi, mediante il quale il professionista deve cercare di risolvere le diverse situazioni di pratica professionale che gli si presentano durante il programma. Lo studente potrà usufruire di un innovativo sistema di video interattivi creati da esperti di rinomata fama.



02

Perché iscriversi a questo Master Semipresenziale?

La domanda di Sviluppo di Software è in costante crescita, a causa del progresso tecnologico. Infatti, gli esperti prevedono che questo profilo professionale crescerà del 30% nei prossimi anni. Di fronte a questo, gli informatici hanno bisogno di incorporare nella loro pratica le metodologie più innovative per lo sviluppo di applicazioni in rete. Ecco perché TECH ha creato questo titolo pionieristico, in cui si combina l'aggiornamento più recente in settori come l'ingegneria del software o la creazione di algoritmi con un tirocinio educativo pratico in un'entità di riferimento. In questo modo, gli studenti acquisiranno competenze avanzate per offrire servizi di alta qualità.

Un piano di studi ad alta intensità che porrà le basi per la tua crescita professionale e ti porterà al vertice dell'informatica"

1. Aggiornarsi sulla base delle più recenti tecnologie

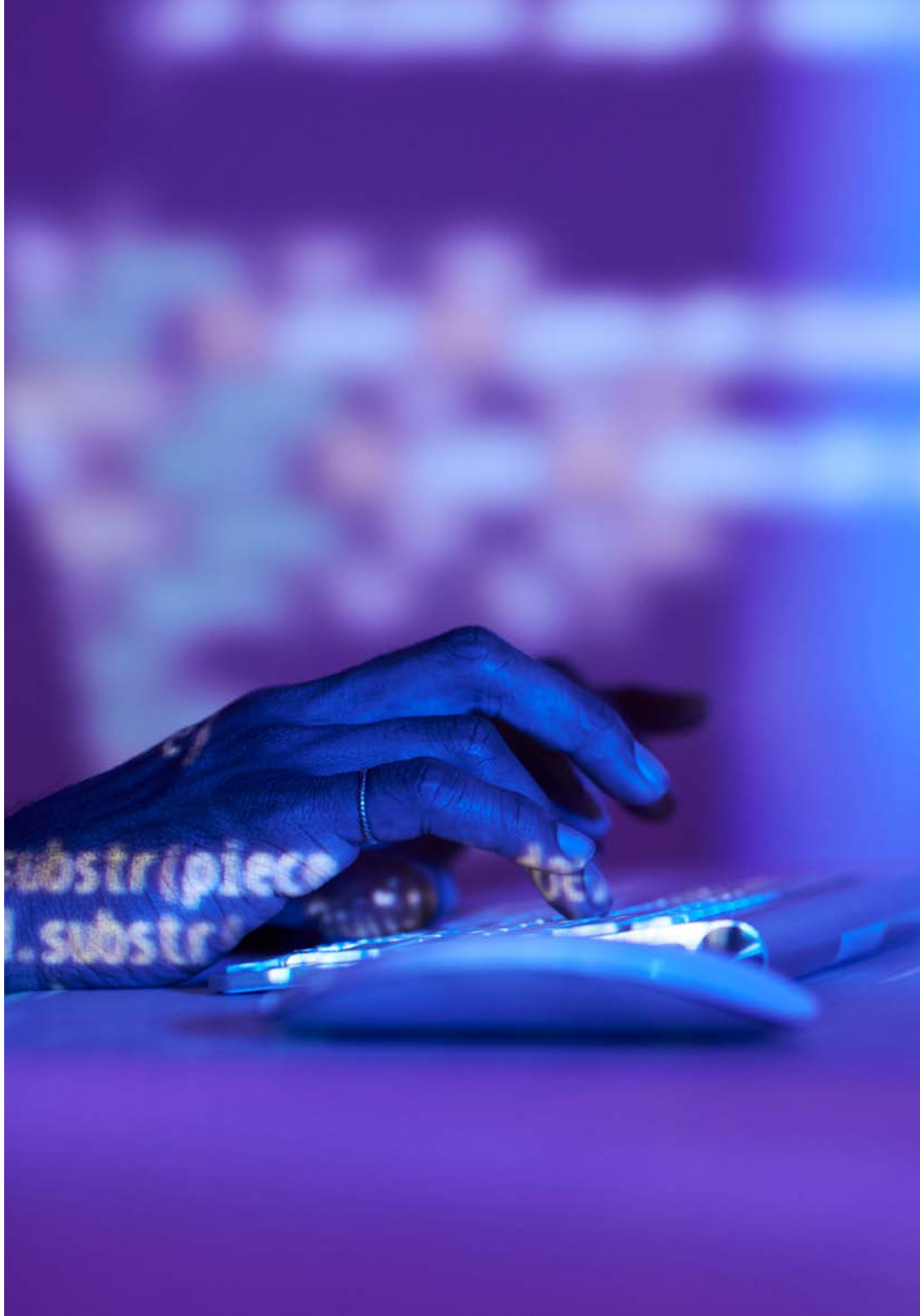
Le nuove tecnologie stanno trasformando in modo significativo lo Sviluppo di Software, migliorandone la produttività, l'efficienza e la qualità. Questi strumenti consentono agli Informatici di affrontare sfide più complesse, creare applicazioni più robuste e adattarsi rapidamente alle mutevoli esigenze del mercato. In questo contesto, TECH presenta questo Master Semipresenziale, che metterà a disposizione degli studenti gli strumenti più sofisticati per svolgere il loro lavoro con efficacia.

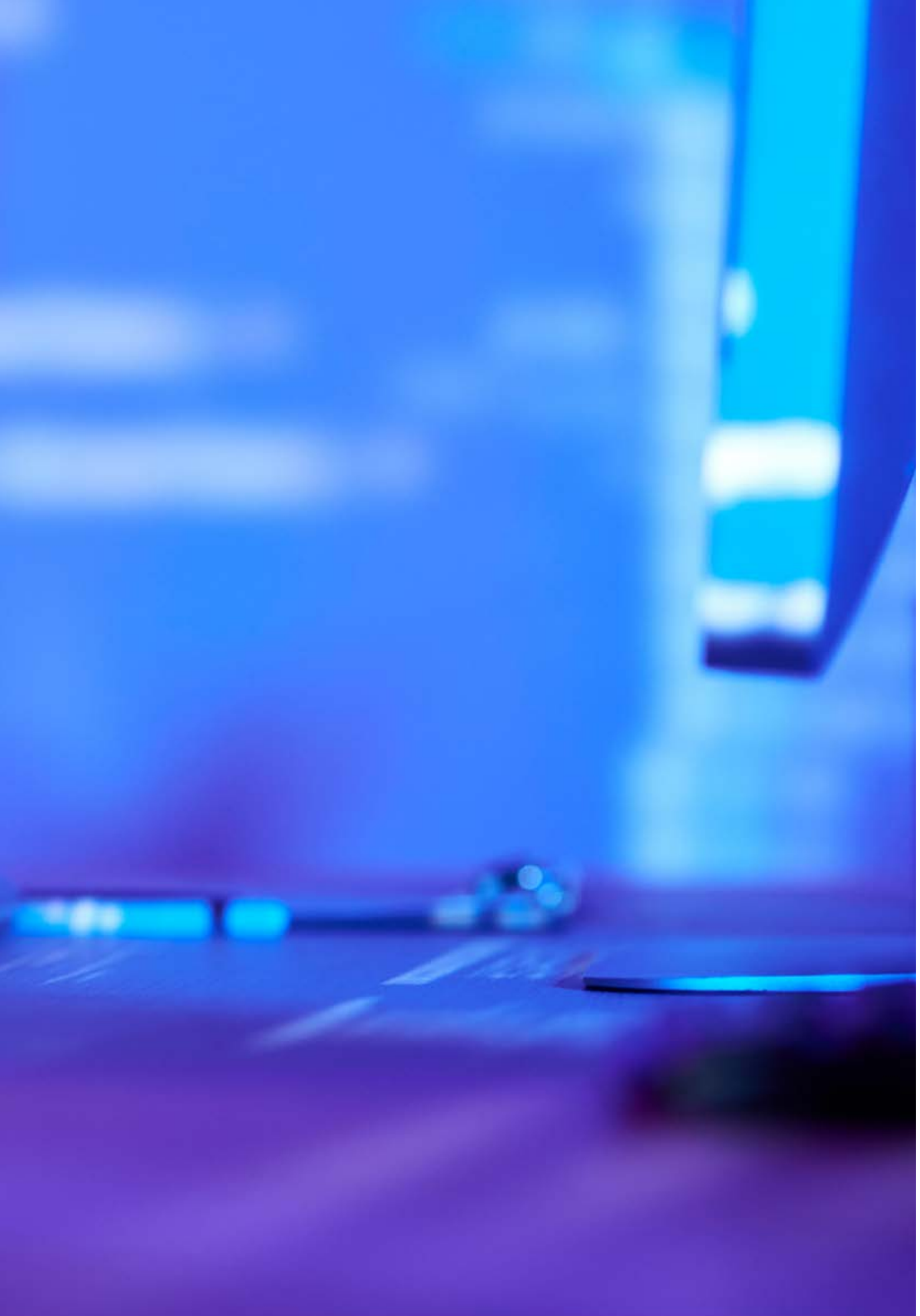
2. Approfondire nuove competenze dall'esperienza dei migliori specialisti

Durante tutto il periodo di pratica, un team di professionisti dello sviluppo software accompagnerà gli studenti per aiutarli a trarre il massimo vantaggio da questa esperienza accademica. Allo stesso tempo, ti trasmetteranno le tecniche più innovative per creare le architetture software più complete e accessibili.

3. Accedere ad ambienti professionali di prim'ordine

La massima premessa di TECH è quella di mettere a disposizione di chiunque programmi universitari di prima classe. Per questo motivo, sceglie con cura tutti i centri disponibili per la realizzazione dei Tirocini. Grazie a questo sforzo, gli informatici avranno accesso a istituzioni di riferimento nel campo dello Sviluppo di Software. In questo modo, saranno in grado di verificare quotidianamente un'area di lavoro esigente, rigorosa ed esaustiva, applicando sempre le ultime tecniche alla propria metodologia di lavoro.





4. Combinare la migliore teoria con la pratica più avanzata

Il mercato accademico è pieno di titoli pedagogici che si limitano a fornire contenuti teorici, dimenticando che la pratica è un aspetto fondamentale per gli studenti nell'applicare le conoscenze alle situazioni reali di lavoro. Lontano da questo, TECH offre un modello di apprendimento 100% pratico, che consentirà agli studenti di acquisire esperienza pratica e affrontare le sfide reali che possono incontrare nella loro carriera professionale.

5. Ampliare le frontiere della conoscenza

TECH offre l'opportunità di svolgere questo Master Semipresenziale in istituti di portata internazionale. Grazie a questo, gli informatici saranno in grado di espandere i loro confini e raggiungere i migliori professionisti che esercitano nel loro lavoro in aziende di primo livello. Un'opportunità unica che solo TECH, la più grande università digitale del mondo, può offrire.

Avrai un'immersione pratica totale nel centro che tu stesso scegli"

03

Obiettivi

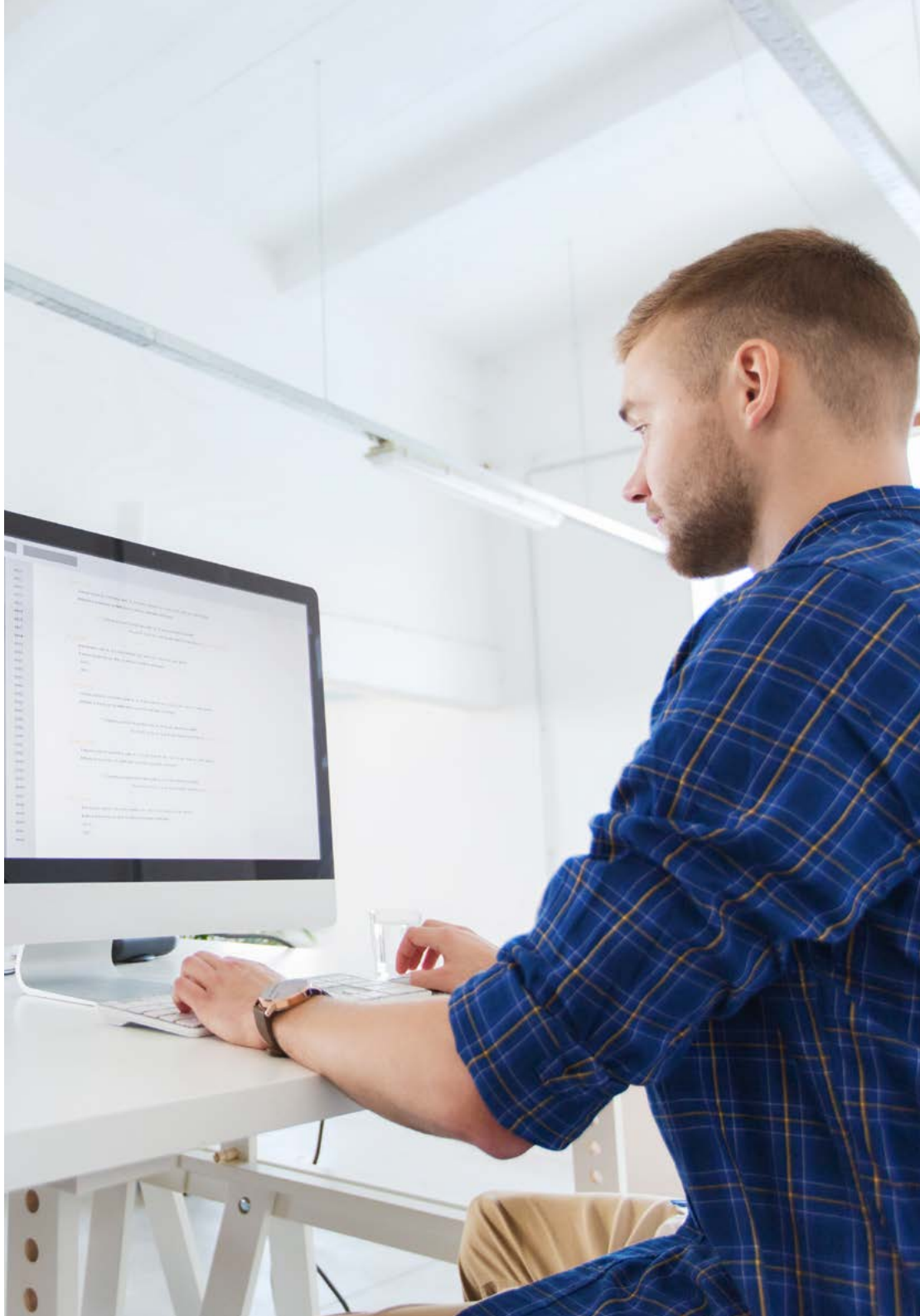
Grazie a questo titolo universitario, gli informatici padroneggeranno linguaggi di programmazione moderni come il C++. Acquisiranno anche le competenze per scrivere codici puliti, efficienti e facili da mantenere. In questo modo, gli sviluppatori progetteranno architetture Software che supportano la scalabilità, la flessibilità e l'integrità dei sistemi.

Questo programma universitario ti fornirà le strategie più all'avanguardia per creare Database altamente efficienti"

Obiettivo generale

- Il presente Master Semipresenziale in Sviluppo di Software fornirà agli informatici sia le conoscenze che le competenze pratiche necessarie per affrontare le sfide in questo settore. I laureati applicheranno efficacemente gli standard di progettazione per risolvere problemi comuni e costruire soluzioni informatiche robuste. Saranno inoltre in grado di mitigare le vulnerabilità dei programmi attraverso l'implementazione di pratiche di sviluppo sicuro. Inoltre, gli studenti creeranno e gestiranno database relazionali per soddisfare le esigenze di archiviazione e recupero delle informazioni

Raggiungi il tuo massimo potenziale nel campo dello Sviluppo di Software grazie ai materiali didattici più completi del mercato accademico"



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 sulla gestione dei file
- Comprendere le diverse tecniche di test del software e l'importanza di generare una buona documentazione insieme a un buon codice sorgente
- Imparare le basi 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 di priorità
- 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
- 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. 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
- Approfondire 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 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. 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 il metodo della secante

Modulo 7. 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 8. 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 di programmazione concorrente, i processi e 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 9. 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* per Front-End Bootstrap per la progettazione di siti web
- Comprendere la struttura del Model-View-Controller nello sviluppo di siti web dinamici
- Conoscere l'architettura orientata ai servizi e le basi del protocollo HTTP

Modulo 10. Ingegneria del Software

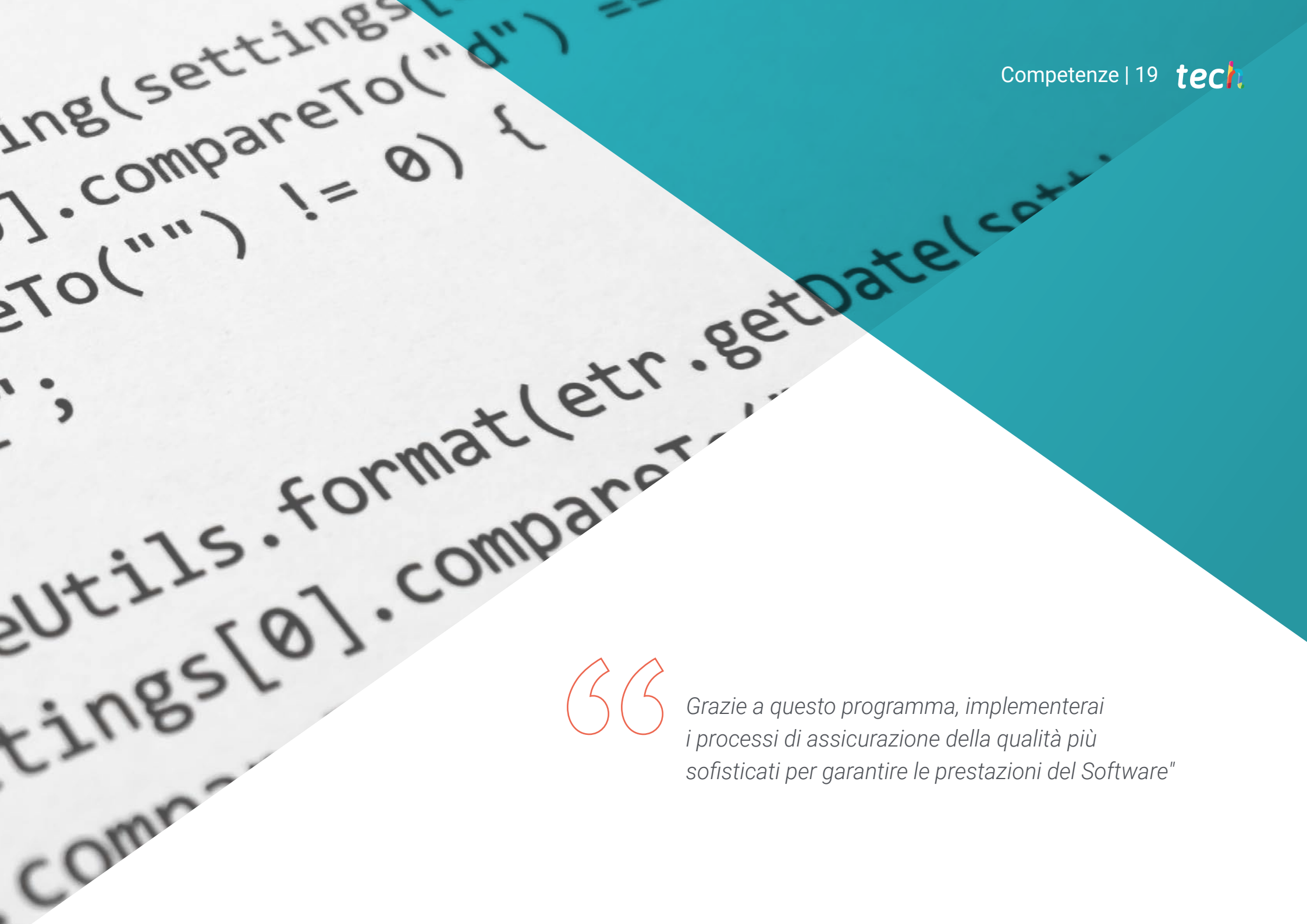
- Porre le basi dell'ingegneria del Software e della modellazione, apprendendo i principali processi e concetti
- Comprendere il processo del Software e i diversi modelli di sviluppo del software, comprese le tecnologie agili
- Comprendere l'ingegneria dei requisiti, il loro sviluppo, la elaborazione, la negoziazione e la convalida
- Imparare la modellazione dei requisiti e i diversi elementi come scenari, informazioni, classi di analisi, flussi, comportamenti e modelli
- Comprendere i concetti e i processi di progettazione del software, apprendendo anche la progettazione dell'architettura, la progettazione a livello di componenti e la progettazione basata su pattern
- Conoscere i principali standard relativi alla qualità del software e alla gestione dei progetti

Combinerai teoria e pratica professionale attraverso un approccio educativo esigente e gratificante"

04

Competenze

Al termine di questo titolo universitario, gli informatici acquisiranno competenze avanzate per superare le sfide nel campo dello Sviluppo di Software. Allo stesso modo, gli studenti padroneggeranno linguaggi di programmazione come C++ che permetteranno loro di creare applicazioni ad alte prestazioni. Gli sviluppatori implementeranno metodologie agili (come Scrum) nei loro progetti per ottimizzare la flessibilità e la reattività del software.



Grazie a questo programma, implementerai i processi di assicurazione della qualità più sofisticati per garantire le prestazioni del Software"

Competenze generali

- Rispondere alle esigenze attuali del settore dello sviluppo Software
- Essere in grado di comprendere la struttura di base di un computer, Software e linguaggi di programmazione per scopi generali
- Applicare i fondamenti della programmazione in C++ includendo classi, variabili, espressioni condizionali e oggetti
- Approfondire le principali strategie di progettazione degli algoritmi, nonché i diversi metodi e misure per il loro calcolo

Questo titolo universitario ti offre l'opportunità di ampliare le tue conoscenze in situazioni reali, con il massimo rigore scientifico di un'istituzione all'avanguardia tecnologica"

Competenze specifiche

- Conoscere le diverse applicazioni e finalità dei sistemi di database, nonché il loro funzionamento e architettura, e applicarli nella vita quotidiana
- Essere in grado di introdurre i diversi sistemi di database attualmente sul mercato
- Saper analizzare algoritmi ricorsivi e di tipo Divide et impera, oltre che effettuare analisi ammortizzate
- Utilizzare la conoscenza dell'interazione uomo-computer e la creazione di interfacce utilizzabili nell'esercizio quotidiano della professione
- Approfondire le conoscenze di programmazione
- Imparare le caratteristiche del linguaggio di markup HTML e il suo utilizzo nella creazione di siti web insieme ai fogli di stile CSS

05

Struttura e contenuti

I materiali didattici che compongono questo Master Semipresenziale sono stati elaborati da veri esperti nel campo dello Sviluppo di Software. Composto da 10 moduli specializzati, il programma doterà gli informatici di una vasta gamma di competenze tecniche. Il programma approfondirà aspetti come Progettazione di Algoritmi, Creazione di Database o Ingegneria del Software. Inoltre, il programma fornirà agli sviluppatori le tecniche più innovative per l'ottimizzazione delle risorse, tra cui il *Backtracking*. In questo modo, gli studenti acquisiranno competenze avanzate per progettare architetture di Software che supportano la crescita e l'evoluzione dei sistemi.

Padroneggia metodologie agili come Scrum per migliorare l'efficienza nello Sviluppo di Software"

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
- 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 ricorsivaiterativa
- 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. ADT: Elenco 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. Heap e Coda di priorità
- 2.6.1. Insiemi di ADT
- 2.6.2. Coda prioritaria ADT

2.7. Tabelle Hash
- 2.7.1. ADT: Tabella Hash
- 2.7.2. Funzioni Hash
- 2.7.3. Funzione 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 id grafi
- 2.8.3. Rappresentazione grafica e operazioni di base
- 2.8.4. Progettazione dei grafi

2.9. Algoritmi e concetti avanzati sui Grafi
- 2.9.1. Problemi dei Grafi
- 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 mix (*Merge Sort*)
- 3.3.6. Ordinamento rapido (*Quicksort*)

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

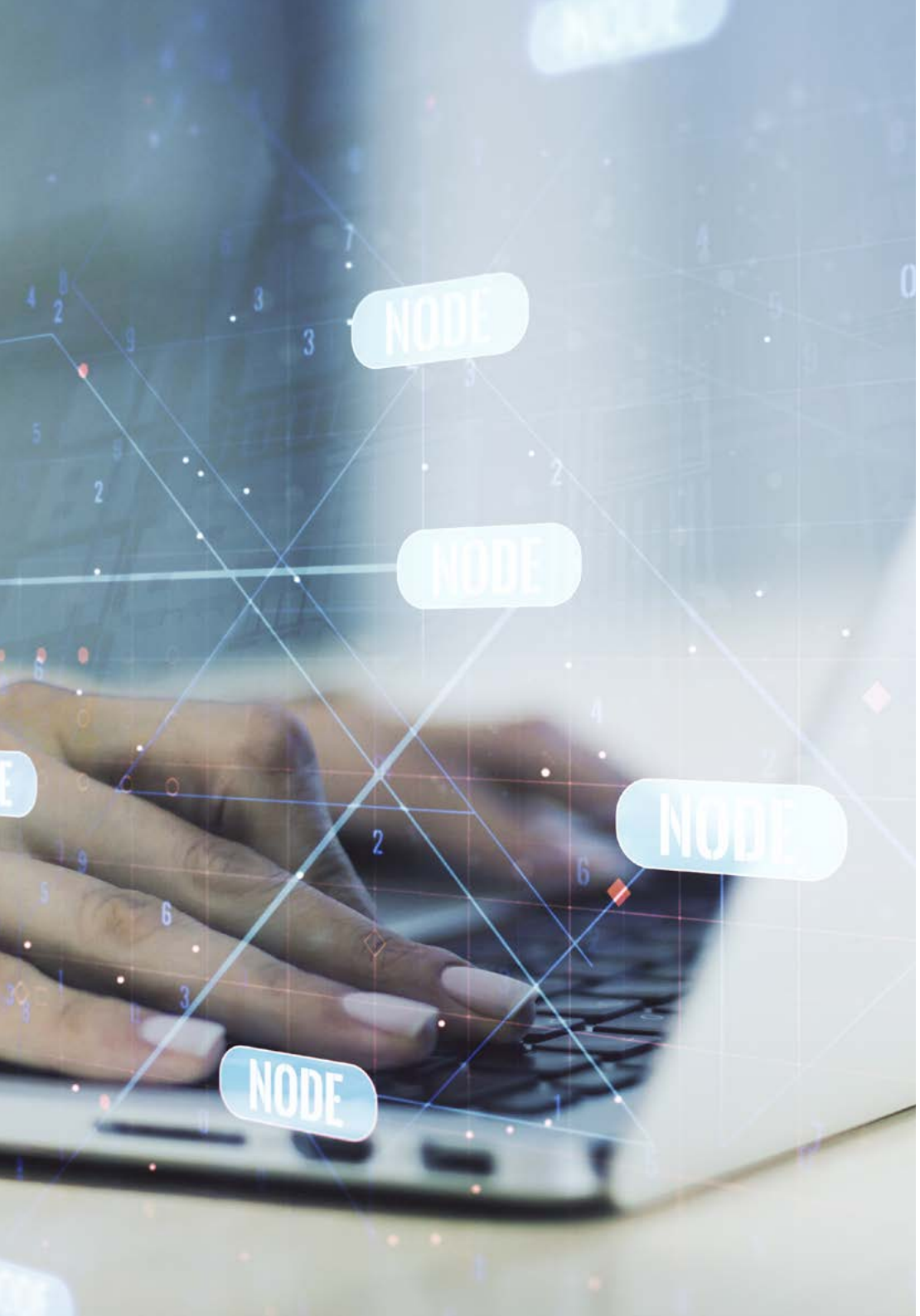
3.6. Algoritmi con grafi
- 3.6.1. Rappresentazione
- 3.6.2. Percorso in larghezza
- 3.6.3. Percorso in profondità
- 3.6.4. Ordinamento topologico

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



Modulo 4. Database

4.1. Applicazioni e scopi dei sistemi di database
- 4.1.1. Applicazioni di diversi sistemi di database
- 4.1.2. Scopo dei diversi sistemi di database
- 4.1.3. Visione dei dati

4.2. Database e architettura
- 4.2.1. Database relazionale
- 4.2.2. Progettazione di database
- 4.2.3. Database a oggetti e semi-strutturati
- 4.2.4. Memorizzazione dei dati e interrogazioni
- 4.2.5. Gestione delle transazioni
- 4.2.6. Estrazione e analisi dei dati
- 4.2.7. Architettura del database

4.3. Il modello relazionale: struttura, operazioni e algebra relazionale estesa
- 4.3.1. La struttura dei database relazionali
- 4.3.2. Operazioni fondamentali dell'algebra relazionale
- 4.3.3. Altre operazioni dell'algebra relazionale
- 4.3.4. Operazioni di algebra relazionale estesa
- 4.3.5. Valori nulli
- 4.3.6. Modifica del database

4.4. SQL (I)
- 4.4.1. Cos'è SQL?
- 4.4.2. La definizione di dati
- 4.4.3. La struttura di base delle query SQL
- 4.4.4. Operazioni sugli insiemi
- 4.4.5. Funzioni di aggregazione
- 4.4.6. Valori nulli

4.5. SQL (II)
- 4.5.1. Subquery annidate
- 4.5.2. Query complesse
- 4.5.3. Visualizzazioni
- 4.5.4. Cursori
- 4.5.5. Query complesse
- 4.5.6. Trigger

4.6. Progettazione di database e modello E-R
- 4.6.1. Panoramica del processo di progettazione
- 4.6.2. Modello entitàrelazione
- 4.6.3. Restrizioni

4.7. Diagrammi entitàrelazione
- 4.7.1. Diagrammi entitàrelazione
- 4.7.2. Aspetti di progettazione delle entità delle relazioni
- 4.7.3. Insiemi di entità deboli

4.8. Il modello esteso di entitàrelazione
- 4.8.1. Caratteristiche del modello E-R esteso
- 4.8.2. Progettazione del database
- 4.8.3. Riduzione a schemi relazionali

4.9. Progettazione di database relazionali
- 4.9.1. Caratteristiche di un buon progetto relazionale
- 4.9.2. Domini atomici e prima forma normale (1FN)
- 4.9.3. Decomposizione mediante dipendenze funzionali
- 4.9.4. Teoria della dipendenza funzionale
- 4.9.5. Algoritmi di decomposizione
- 4.9.6. Decomposizione con dipendenze multivariate
- 4.9.7. Altre forme normali
- 4.9.8. Processo di progettazione del database

4.10. Database NoSQL
- 4.10.1. Cosa sono i database NoSQL?
- 4.10.2. Analisi delle diverse opzioni NoSQL e delle loro caratteristiche
- 4.10.3. MongoDB

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
- 5.3.4. Parallelismo delle query
- 5.3.5. Parallelismo tra query
- 5.3.6. Progettazione di sistemi paralleli
- 5.3.7. 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 ai grafi
- 5.9.3. Database chiave-valore

5.10. Data Warehouse, OLAP e estrazione dei dati
- 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. Progettazione avanzata degli algoritmi

6.1. Analisi di algoritmi ricorsivi e divide et impera
- 6.1.1. Porre e risolvere equazioni di ricorrenza omogenee e non omogenee
- 6.1.2. Panoramica della strategia divide et impera

6.2. Analisi ammortizzata
- 6.2.1. Analisi aggregata
- 6.2.2. Il metodo di contabilizzazione
- 6.2.3. Il metodo del potenziale

6.3. Programmazione dinamica e algoritmi per problemi NP
- 6.3.1. Caratteristiche della programmazione dinamica
- 6.3.2. Monitoraggio a ritroso: *Backtracking*
- 6.3.3. Ramificazione e potatura

6.4. Ottimizzazione combinatoria
- 6.4.1. Rappresentazione del problema
- 6.4.2. Ottimizzazione 1D

6.5. Algoritmi di randomizzazione
- 6.5.1. Esempi di algoritmi di randomizzazione
- 6.5.2. Il teorema di Buffon
- 6.5.3. Algoritmo di Monte Carlo
- 6.5.4. Algoritmo di Las Vegas

6.6. Ricerca locale e di candidati
- 6.6.1. *Gradient ascent*
- 6.6.2. *Hill Climbing*
- 6.6.3. *Simulated Annealing*
- 6.6.4. *Tabu Search*
- 6.6.5. Ricerca di candidati

6.7. Verifica formale dei programmi
- 6.7.1. Specifica delle astrazioni funzionali
- 6.7.2. Il linguaggio della logica del primo ordine
- 6.7.3. Sistema formale di Hoare

6.8. Verifica di programmi iterativi
- 6.8.1. Regole del sistema formale di Hoare
- 6.8.2. Concetto di iterazioni invarianti

6.9. Metodi numerici
- 6.9.1. Il metodo della bisezione
- 6.9.2. Il metodo Newton Raphson
- 6.9.3. Il metodo della secante

6.10. Algoritmi paralleli
- 6.10.1. Operazioni binarie parallele
- 6.10.2. Operazioni in parallelo con i grafi
- 6.10.3. Parallelismo nel divide et impera
- 6.10.4. Parallelismo nella programmazione dinamica

Modulo 7. Interazione Persona-Computer

7.1. Introduzione all'interazione persona-computer
- 7.1.1. Che cos'è l'interazione persona-computer
- 7.1.2. Rapporto tra interazione persona-computer con altre discipline
- 7.1.3. L'interfaccia utente
- 7.1.4. Usabilità e accessibilità
- 7.1.5. Esperienza utente e design incentrato sull'utente

7.2. Il computer e l'interazione: interfaccia utente e paradigmi di interazione
- 7.2.1. L'interazione
- 7.2.2. Paradigmi di interazione e stili di interazione
- 7.2.3. Evoluzione delle interfacce utente
- 7.2.4. Interfacce utente classica: WIMP/GUI, comandi, voce, realtà virtuale
- 7.2.5. Interfacce utente innovative: mobile, portatile, collaborativa, collaborativa, BCI

7.3. Il fattore umano: aspetti psicologici e cognitivi
- 7.3.1. L'importanza del fattore umano nell'interazione
- 7.3.2. Elaborazione dell'informazione umana
- 7.3.3. L'ingresso e l'uscita delle informazioni: visive, uditive e tattili
- 7.3.4. Percezione e attenzione
- 7.3.5. Conoscenza e modelli mentali: rappresentazione, organizzazione e acquisizione

7.4. Il fattore umano: limitazioni sensoriali e fisiche
- 7.4.1. Diversità funzionale, disabilità e deficit
- 7.4.2. Diversità visiva
- 7.4.3. Diversità dell'udito
- 7.4.4. Diversità cognitiva
- 7.4.5. Diversità motoria
- 7.4.6. Il caso degli immigrati digitali

7.5. Il processo di progettazione (I): analisi dei requisiti per la progettazione dell'interfaccia utente
- 7.5.1. Progettazione incentrata sull'utente
- 7.5.2. Che cos'è l'analisi dei requisiti?
- 7.5.3. Raccolta di informazioni
- 7.5.4. Analisi e interpretazione delle informazioni
- 7.5.5. Analisi di usabilità e accessibilità

7.6. Il processo di progettazione (II): prototipazione e analisi del compito
- 7.6.1. Progetto concettuale
- 7.6.2. Prototipazione
- 7.6.3. Analisi gerarchica dei compiti

7.7. Il processo di progettazione (III): la valutazione
- 7.7.1. La valutazione nel processo di progettazione: obiettivi e metodi
- 7.7.2. Metodi di valutazione senza utenti
- 7.7.3. Metodi di valutazione con gli utenti
- 7.7.4. Standard e norme di valutazione

7.8. Accessibilità: definizione e linee guida
- 7.8.1. Accessibilità e design universale
- 7.8.2. L'iniziativa WAI e le linee guida WCAG
- 7.8.3. Linee guida WCAG 2.0 e 2.1

7.9. Accessibilità: valutazione e diversità funzionale
- 7.9.1. Strumenti di valutazione dell'accessibilità del web
- 7.9.2. Accessibilità e diversità funzionale

7.10. Il computer e l'interazione: periferiche e dispositivi
- 7.10.1. Dispositivi e periferiche tradizionali
- 7.10.2. Dispositivi e periferiche alternative
- 7.10.3. Telefoni cellulari e tablet
- 7.10.4. Diversità funzionale, interazione e periferiche

Modulo 8. Programmazione avanzata

8.1. Introduzione alla programmazione orientata agli oggetti
- 8.1.1. Introduzione alla programmazione orientata agli oggetti
- 8.1.2. Progettazione delle lezioni
- 8.1.3. Introduzione a UML per la modellazione dei problemi

8.2. Relazioni tra lezioni
- 8.2.1. Astrazione ed ereditarietà
- 8.2.2. Concetti avanzati di ereditarietà
- 8.2.3. Polimorfismo
- 8.2.4. Composizione e aggregazione

8.3. Introduzione ai design pattern per i problemi orientati agli oggetti
- 8.3.1. Cosa sono i design pattern
- 8.3.2. Pattern *Factory*
- 8.3.4. Pattern *Singleton*
- 8.3.5. Pattern *Observer*
- 8.3.6. Pattern *Composite*

8.4. Eccezioni
- 8.4.1. Quali sono le eccezioni?
- 8.4.2. Gestione e acquisizione delle eccezioni
- 8.4.3. Avvio delle eccezioni
- 8.4.4. Creazione di eccezioni

8.5. Interfacce utente
- 8.5.1. Introduzione a Qt
- 8.5.2. Posizionamento
- 8.5.3. Cosa sono gli eventi?
- 8.5.4. Eventi: definizione e acquisizione
- 8.5.5. Sviluppo di interfacce utente

8.6. Introduzione alla programmazione concorrente
- 8.6.1. Introduzione alla programmazione concorrente
- 8.6.2. Il concetto di processo e di thread
- 8.6.3. Interazione tra processi o thread
- 8.6.4. Thread in C++
- 8.6.6. Vantaggi e svantaggi della programmazione concorrente

8.7. Gestione e sincronizzazione dei thread
- 8.7.1. Ciclo di vita di un thread
- 8.7.2. La classe *Thread*
- 8.7.3. Pianificazione del thread
- 8.7.4. Gruppi di thread
- 8.7.5. Thread di tipo demoniaco
- 8.7.6. Sincronizzazione
- 8.7.7. Meccanismi di bloccaggio
- 8.7.8. Meccanismi di comunicazione
- 8.7.9. Monitor

8.8. Problemi comuni nella programmazione concorrente
- 8.8.1. Il problema dei produttori-consumatori
- 8.8.2. Il problema dei lettori e degli scrittori
- 8.8.3. Il problema della cena dei filosofi

8.9. Documentazione e test del software
- 8.9.1. Perché è importante documentare il software?
- 8.9.2. Documentazione di progettazione
- 8.9.3. Utilizzo di strumenti per la documentazione

8.10. Test di software
- 8.10.1. Introduzione al test del software
- 8.10.2. Tipi di test
- 8.10.3. Test dell'unità
- 8.10.4. Test di integrità
- 8.10.5. Test di convalida
- 8.10.6. Test del sistema

Modulo 9. Sviluppo delle applicazioni in rete

9.1. Linguaggi di mercato HTML5
- 9.1.1. Nozioni di base sulla HTML
- 9.1.2. Nuovi elementi HTML5
- 9.1.3. Moduli: nuovi controlli

9.2. Introduzione ai fogli di stile CSS
- 9.2.1. Primi passi con CSS
- 9.2.2. Introduzione ai CSS3

9.3. Linguaggio di scripting del browser: JavaScript
- 9.3.1. Nozioni di base di JavaScript
- 9.3.2. DOM
- 9.3.3. Eventi
- 9.3.4. JQuery
- 9.3.5. Ajax

9.4. Concetto di programmazione orientata ai componenti
- 9.4.1. Contesto
- 9.4.2. Componenti e interfacce
- 9.4.3. Stati di un componente

9.5. Architettura dei componenti
- 9.5.1. Architetture attuali
- 9.5.2. Integrazione e distribuzione dei componenti

9.6. *Framework Frontend*: Bootstrap
- 9.6.1. Design con la rete
- 9.6.2. Formulari
- 9.6.3. Componenti

9.7. Controllore della vista del modello
- 9.7.1. Metodi di sviluppo web
- 9.7.2. Pattern di progettazione: MVC

9.8. Tecnologie Grid informative
- 9.8.1. Aumento delle risorse informatiche
- 9.8.2. Concetto di tecnologia Grid

9.9. Architetture orientate ai servizi
- 9.9.1. SOA e servizi web
- 9.9.2. Topologia del servizio web
- 9.9.3. Piattaforme di servizi web

9.10. Protocollo HTTP
- 9.10.1. Messaggi
- 9.10.2. Sessioni persistenti
- 9.10.3. Sistema crittografico
- 9.10.4. Funzionamento del protocollo HTTPS

Modulo 10. Ingegneria del Software

10.1. Introduzione all'ingegneria del Software e alla modellazione
- 10.1.1. La natura del software
- 10.1.2. La natura unica delle Webapp
- 10.1.3. Ingegneria del Software
- 10.1.4. Il processo del Software
- 10.1.5. La pratica dell'ingegneria del Software
- 10.1.6. Miti del Software
- 10.1.7. Come tutto ha inizio
- 10.1.8. Concetti orientati agli oggetti
- 10.1.9. Introduzione a UML

10.2. Il processo del Software
- 10.2.1. Un modello generale di processo
- 10.2.2. Modelli di processo prescrittivi
- 10.2.3. Modelli di processo specializzati
- 10.2.4. Il processo unificato
- 10.2.5. Modelli di processo personali e di gruppo
- 10.2.6. Che cos'è l'agilità?
- 10.2.7. Che cos'è un processo agile?
- 10.2.8. Scrum
- 10.2.9. Kit di strumenti per i processi agili

10.3. Principi che guidano la pratica dell'ingegneria del software
- 10.3.1. Principi che guidano il processo
- 10.3.2. Principi che guidano la pratica
- 10.3.3. Principi di comunicazione
- 10.3.4. Principi di pianificazione
- 10.3.5. Principi di modellazione
- 10.3.6. Principi di costruzione
- 10.3.7. Principi di implementazione

10.4. Comprendere i requisiti
- 10.4.1. Ingegneria dei requisiti
- 10.4.2. Porre le basi
- 10.4.3. Indagine sui requisiti
- 10.4.4. Sviluppo di casi d'uso
- 10.4.5. Elaborazione del modello dei requisiti
- 10.4.6. Negoziazione dei requisiti
- 10.4.7. Convalida dei requisiti

10.5. Modellazione dei requisiti: scenari, informazioni e classi di analisi
- 10.5.1. Analisi dei requisiti
- 10.5.2. Modellazione basata su scenari
- 10.5.3. Modelli UML che forniscono il caso d'uso
- 10.5.4. Concetti di modellazione dei dati
- 10.5.5. Modellazione basata sulle classi
- 10.5.6. Diagrammi di classe

10.6. Modellazione dei requisiti: flusso, comportamento e modelli
- 10.6.1. Strategie di definizione dei requisiti
- 10.6.2. Modellazione orientata al flusso
- 10.6.3. Diagrammi di stato
- 10.6.4. Creare un modello comportamentale
- 10.6.5. Diagrammi di sequenza
- 10.6.6. Diagrammi di comunicazione
- 10.6.7. Schemi per la modellazione dei requisiti

10.7. Concetti di design
- 10.7.1. Il design nel contesto dell'ingegneria del software
- 10.7.2. Processo di progettazione
- 10.7.3. Concetti di design
- 10.7.4. Concetti di design orientati agli oggetti
- 10.7.5. Il modello di design

10.8. Design architettonico
- 10.8.1. Architettura del software
- 10.8.2. Generi architettonici
- 10.8.3. Stili architettonici
- 10.8.4. Design architettonico
- 10.8.5. Evoluzione dei design alternativi per l'architettura
- 10.8.6. Mappatura dell'architettura con l'uso di flussi di dati

10.9. Design a livello di componente e basato su pattern
- 10.9.1. Che cos'è un componente?
- 10.9.2. Design dei componenti basato sulle classi
- 10.9.3. Realizzazione del progetto a livello di componenti
- 10.9.4. Design dei componenti tradizionali
- 10.9.5. Sviluppo basato su componenti
- 10.9.6. Pattern di progettazione
- 10.9.7. Il design del software basato su modelli
- 10.9.8. Modelli architettonici
- 10.9.9. Modelli di design a livello di componenti
- 10.9.10. Modelli di design dell'interfaccia utente

10.10. Qualità del software e gestione dei progetti
- 10.10.1. Qualità del Software
- 10.10.2. Il dilemma della qualità del software
- 10.10.3. Raggiungere la qualità del software
- 10.10.4. Garanzia di qualità del software
- 10.10.5. Lo spettro amministrativo
- 10.10.6. Il personale
- 10.10.7. Il prodotto
- 10.10.8. Il processo
- 10.10.9. Il progetto
- 10.10.10. Principi e pratiche

06

Tirocinio

Una volta superato il periodo teorico online, questo programma universitario prevede una fase di formazione pratica in un'entità di riferimento. Durante questo periodo, gli studenti saranno supportati da un tutor che li aiuterà a trarre il massimo vantaggio da questa esperienza. Grazie a questo, gli informatici acquisiranno competenze avanzate per sperimentare un notevole salto di qualità nell'esercizio della loro professione.

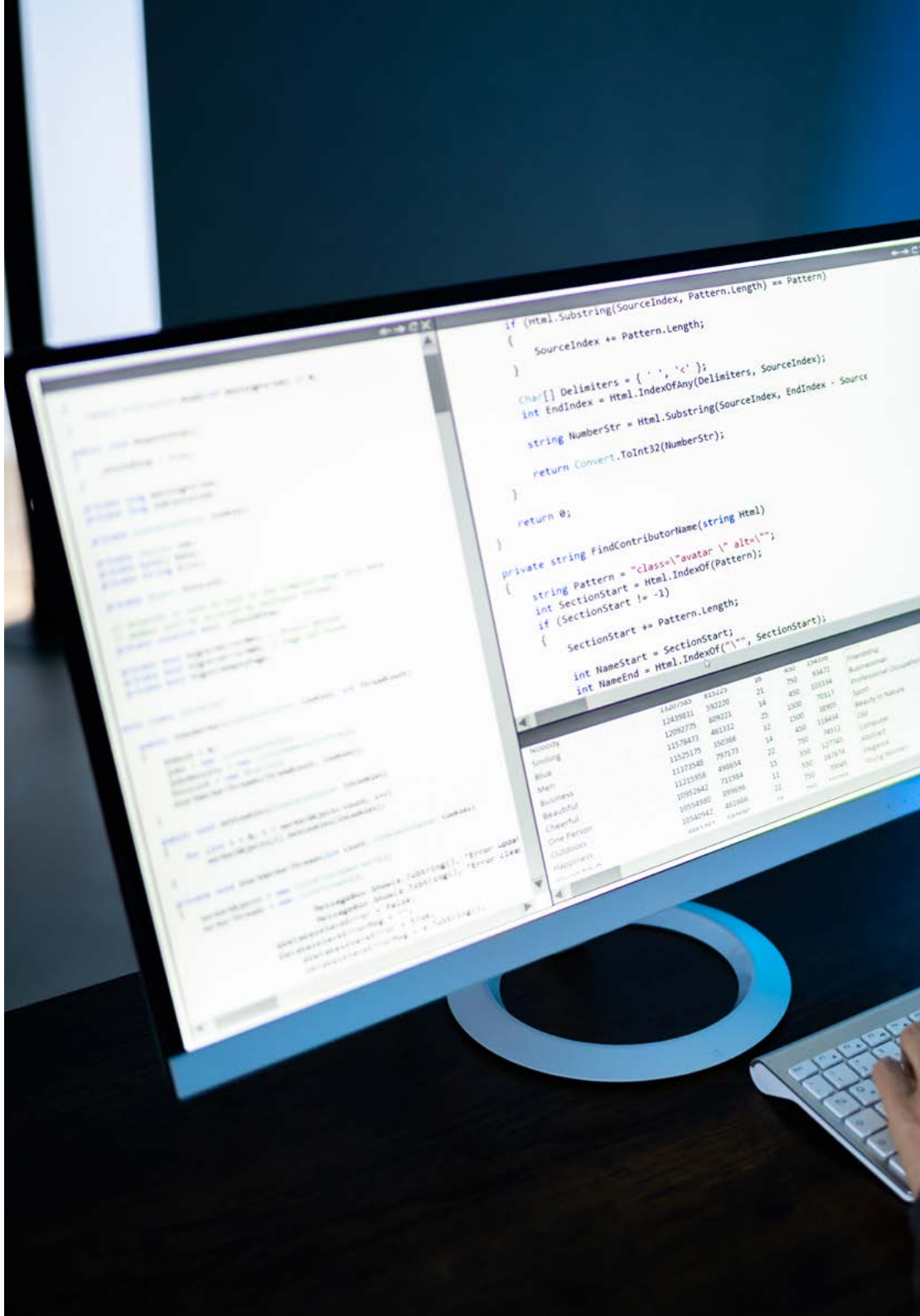
Grazie a questo titolo universitario, sarai preparato per superare le sfide nel campo dello Sviluppo di Software"

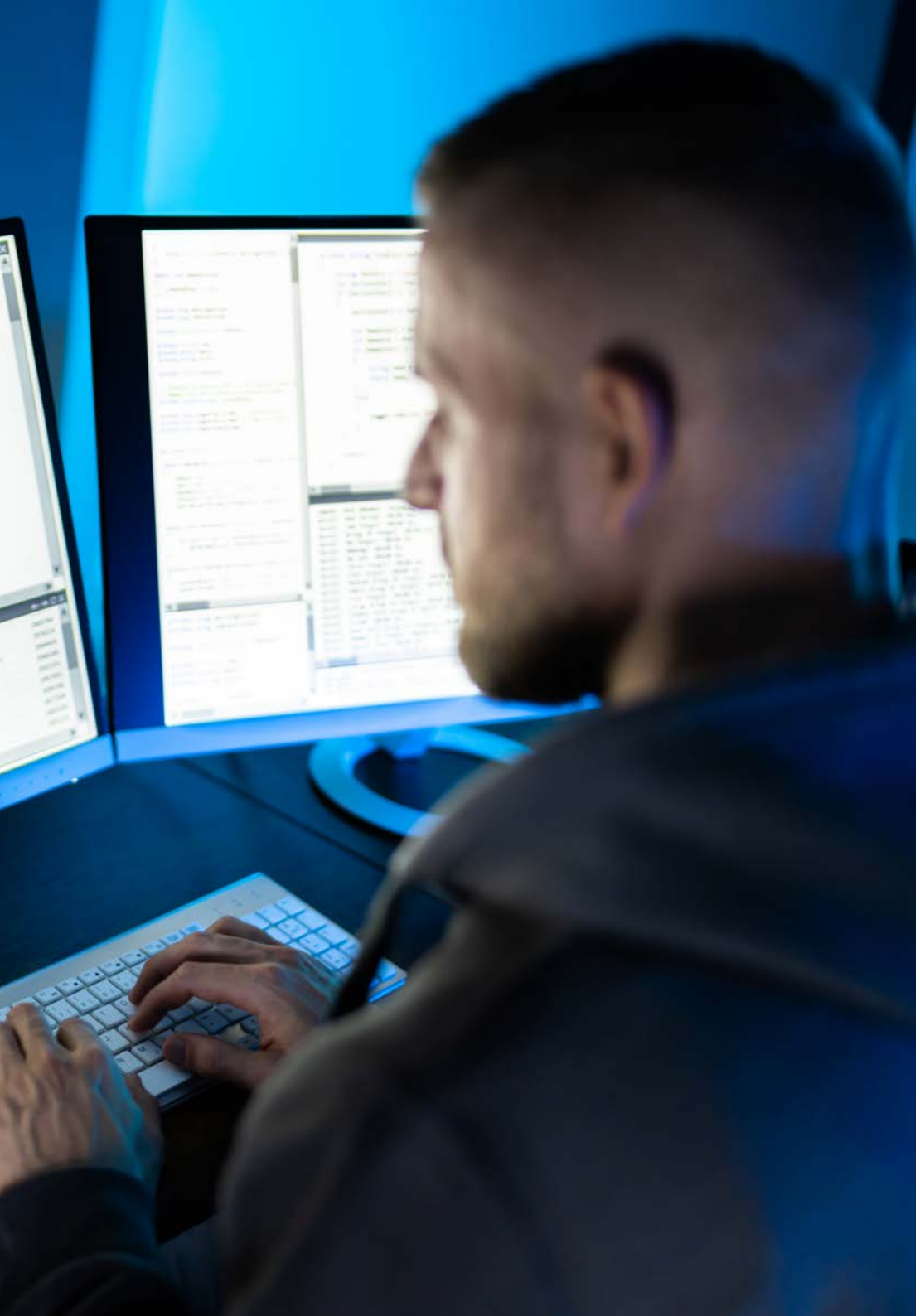
Il Tirocinio di questo programma in Sviluppo di Software è composta da un seminario educativo pratico presso una prestigiosa azienda, Il periodo di durata di 3 settimane, dal lunedì al venerdì e con giornate di 8 ore consecutive di preparazione pratica a fianco di uno specialista strutturato. Questo tirocinio educativo permetterà ai professionisti di applicare i concetti teorici appresi in situazioni reali di lavoro, sia nelle aziende del settore che in progetti collaborativi

In questa proposta di formazione, di carattere completamente pratico, le attività sono dirette allo sviluppo e al perfezionamento delle competenze necessarie per fornire servizi di Sviluppo di Software, nonché condizioni che richiedono un alto livello di qualificazione, orientate alla formazione specifica per l'esercizio dell'attività.

Si tratta senza dubbio di un'eccellente opportunità per gli studenti di immergersi nella realtà di una professione piena di sfide. In questo modo, collaboreranno a progetti relativi alla progettazione di Software, creazione di database o elaborazione di algoritmi, tra gli altri. Così, gli informatici svilupperanno competenze avanzate per ottimizzare la loro pratica e aumentare i loro orizzonti professionali.

L'insegnamento pratico sarà svolto con la partecipazione attiva dello studente svolgendo le attività e le procedure di ogni area di competenza (imparare a imparare e imparare a fare), con l'accompagnamento e la guida di insegnanti e altri compagni di formazione che facilitano il lavoro di squadra e l'integrazione multidisciplinare come competenze trasversali per la pratica dello Sviluppo di Software (imparare a essere e imparare a relazionarsi).





Le procedure descritte di seguito saranno la base della parte pratica della formazione, e la loro realizzazione sarà soggetta alla disponibilità propria del centro e al suo volume di lavoro, essendo le attività proposte come segue:

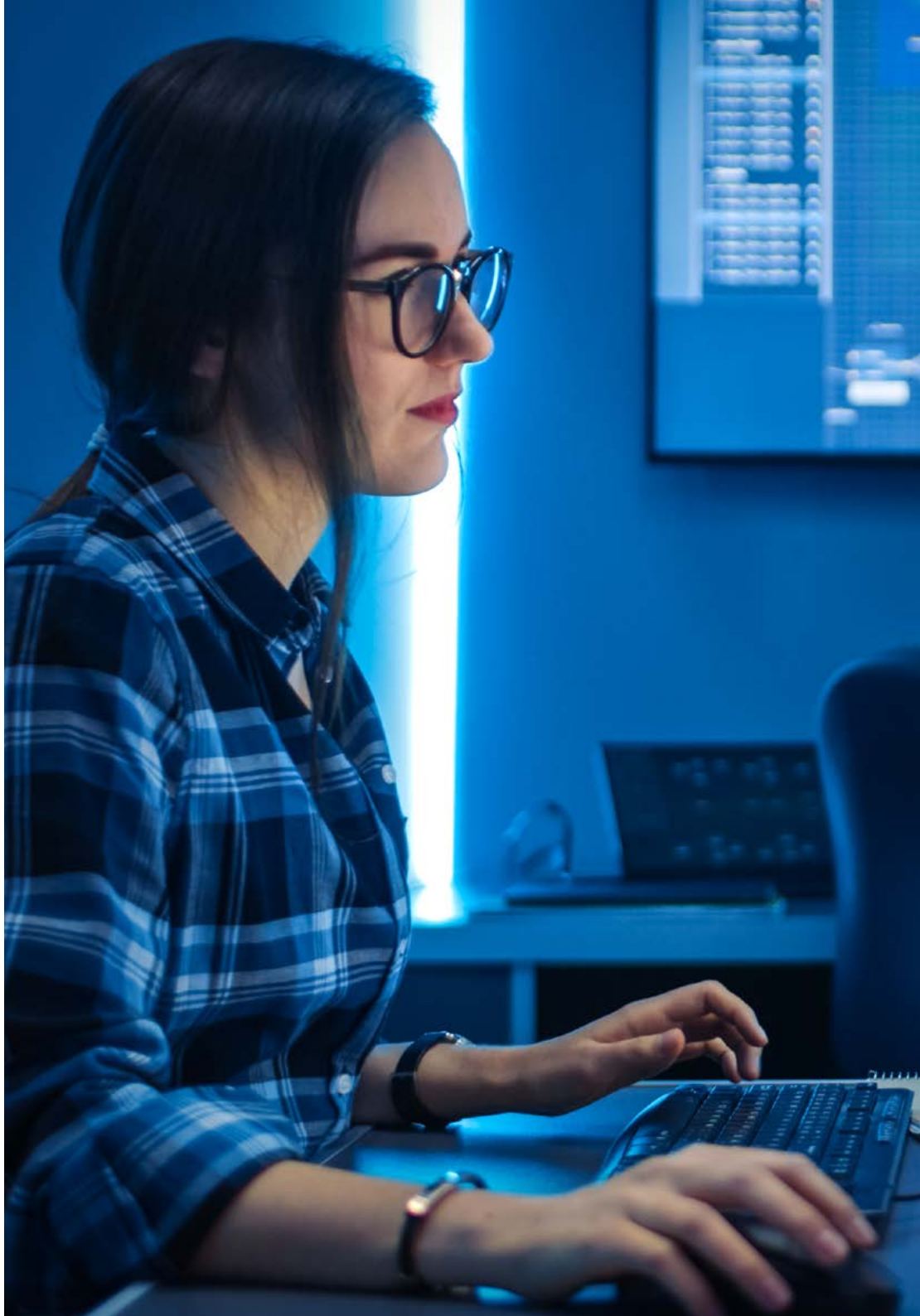
Modulo	Attività Pratica
Architettura di dati	Creare nuove strutture di dati che siano efficienti e adatte a risolvere problemi specifici
	Implementare strutture di dati di base come pule, code, alberi, grafi, ecc.
	Valutare la complessità temporale di diverse strutture algoritmiche
	Realizzare schemi di database efficienti utilizzando sistemi di dati ottimizzare la memorizzazione e il recupero dei dati
Tecniche di Algoritmi	Progettare algoritmi in diversi linguaggi di programmazione
	Utilizzare tecniche avanzate come la programmazione dinamica e algoritmi di grafo
	Sviluppare algoritmi che minimizzano l'impiego di risorse computazionali come memoria e tempo della CPU
	Testare algoritmi per verificarne il corretto funzionamento in diversi scenari e con diversi set di dati
Sistemi di Stoccaggio di Dati	Configurare database in sistemi di gestione come MySQL, PostgreSQL, ecc.
	Utilizzare strumenti di monitoraggio per monitorare le prestazioni del database, al fine di garantire l'affidabilità e la disponibilità
	Applicare controlli di accesso e politiche di sicurezza per proteggere i dati da accessi non autorizzati
	Eseguire la migrazione di database da un ambiente a un altro (ad esempio, da un database locale a uno cloud)
Sviluppo di Software	Eseguire l'architettura del sistema, compresa la divisione in moduli e la specificazione delle interfacce
	Sviluppare disegni dettagliati di ogni componente o modulo del sistema, compresi i diagrammi UML e le specifiche tecniche
	Eseguire test unitari per verificare il corretto funzionamento dei moduli di codice individuali
	Identificare e correggere i bug nel Software dopo la sua distribuzione, per implementare nuove funzionalità o miglioramenti

Assicurazione di responsabilità civile

La preoccupazione principale di questa istituzione è quella di garantire la sicurezza sia dei tirocinanti e degli altri agenti che collaborano ai processi di tirocinio in azienda. All'interno delle misure rivolte a questo fine ultimo, esiste la risposta a qualsiasi incidente che possa verificarsi durante il processo di insegnamento-apprendimento.

A tal fine, questa istituzione educativa si impegna a stipulare un'assicurazione di responsabilità civile per coprire qualsiasi eventualità che possa insorgere durante la permanenza presso il centro di tirocinio.

La polizza di responsabilità civile per i tirocinanti deve garantire una copertura assicurativa completa e deve essere stipulata prima dell'inizio del periodo di tirocinio. Grazie a questa garanzia, il professionista si sentirà privo di ogni tipo di preoccupazione nel caso di eventuali situazioni impreviste che possano sorgere durante il tirocinio e potrà godere di una copertura assicurativa fino al termine dello stesso.



Condizioni generali del tirocinio

Le condizioni generali dell'accordo di tirocinio per il programma sono le seguenti:

1. TUTORAGGIO: durante il Master Semipresenziale agli studenti verranno assegnati due tutor che li seguiranno durante tutto il percorso, risolvendo eventuali dubbi e domande. Da un lato, lo studente disporrà di un tutor professionale appartenente al centro di inserimento lavorativo che lo guiderà e lo supporterà in ogni momento. Dall'altro lato, allo studente verrà assegnato anche un tutor accademico che avrà il compito di coordinare e aiutare lo studente durante l'intero processo, risolvendo i dubbi e fornendogli tutto ciò di cui potrebbe aver bisogno. In questo modo, il professionista sarà accompagnato in ogni momento e potrà risolvere tutti gli eventuali dubbi, sia di natura pratica che accademica.

2. DURATA: il programma del tirocinio avrà una durata di tre settimane consecutive di preparazione pratica, distribuite in giornate di 8 ore lavorative, per cinque giorni alla settimana. I giorni di frequenza e l'orario saranno di competenza del centro, che informerà debitamente e preventivamente il professionista, con un sufficiente anticipo per facilitarne l'organizzazione.

3. MANCATA PRESENTAZIONE: in caso di mancata presentazione il giorno di inizio del Master Semipresenziale, lo studente perderà il diritto allo stesso senza possibilità di rimborso o di modifica di date. L'assenza per più di due giorni senza un giustificato motivo/certificato medico comporterà la rinuncia dello studente al tirocinio e, pertanto, la relativa automatica cessazione. In caso di ulteriori problemi durante lo svolgimento del tirocinio, essi dovranno essere debitamente e urgentemente segnalati al tutor accademico.

4. CERTIFICAZIONE: lo studente che supererà il Master Semipresenziale riceverà un certificato che attesterà il tirocinio svolto presso il centro in questione.

5. RAPPORTO DI LAVORO: il Master Semipresenziale non costituisce alcun tipo di rapporto lavorativo.

6. STUDI PRECEDENTI: alcuni centri potranno richiedere un certificato di studi precedenti per la partecipazione al Master Semipresenziale. In tal caso, sarà necessario esibirlo al dipartimento tirocini di TECH affinché venga confermata l'assegnazione del centro prescelto.

7. NON INCLUDE: il Master Semipresenziale non includerà nessun elemento non menzionato all'interno delle presenti condizioni. Pertanto, non sono inclusi alloggio, trasporto verso la città in cui si svolge il tirocinio, visti o qualsiasi altro servizio non menzionato.

Tuttavia, gli studenti potranno consultare il proprio tutor accademico per qualsiasi dubbio o raccomandazione in merito. Egli fornirà tutte le informazioni necessarie per semplificare le procedure.

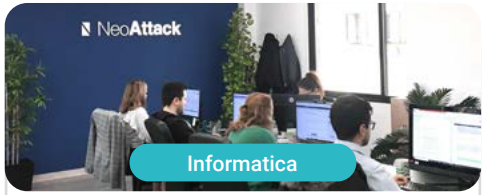
07

Dove posso svolgere il Tirocinio?

In linea con il suo impegno di offrire un'istruzione di alta qualità accessibile a tutti, TECH amplia le opportunità accademiche degli studenti e consente che il tirocinio possa essere svolto presso varie entità di fama internazionale. Così, gli studenti hanno un'opportunità ideale per migliorare la loro qualità professionale lavorando con i migliori specialisti nel campo dello Sviluppo di Software.

Completerai il tuo apprendistato in Sviluppo di Software con un'esperienza pratica insieme a professionisti del settore"

Gli studenti potranno svolgere il tirocinio di questo Master Semipresenziale presso i seguenti centri:



NeoAttack

Paese
Spagna

Città
Madrid

Indirizzo: Calle Santa Engracia 151, Planta 1, 1, Madrid

NeoAttack è leader di mercato nel settore SEO e delle strategie pubblicitarie

Tirocini correlati:
Elaborazione grafica
- Sviluppo di Software

Captia Ingeniería

Paese
Spagna

Città
Madrid

Indirizzo: Av. de las Nieves, 37, Bloque A Planta 1 Oficina E, 28935, Móstoles, Madrid

Società di informatica dedicata a fornire soluzioni tecnologiche avanzate per le industrie

Tirocini correlati:
- Visual Analytics and Big Data
- Sviluppo di Software

Potenzia la tua carriera professionale con un insegnamento olistico, che ti consenta di progredire sia dal punto di vista teorico che pratico"

08

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

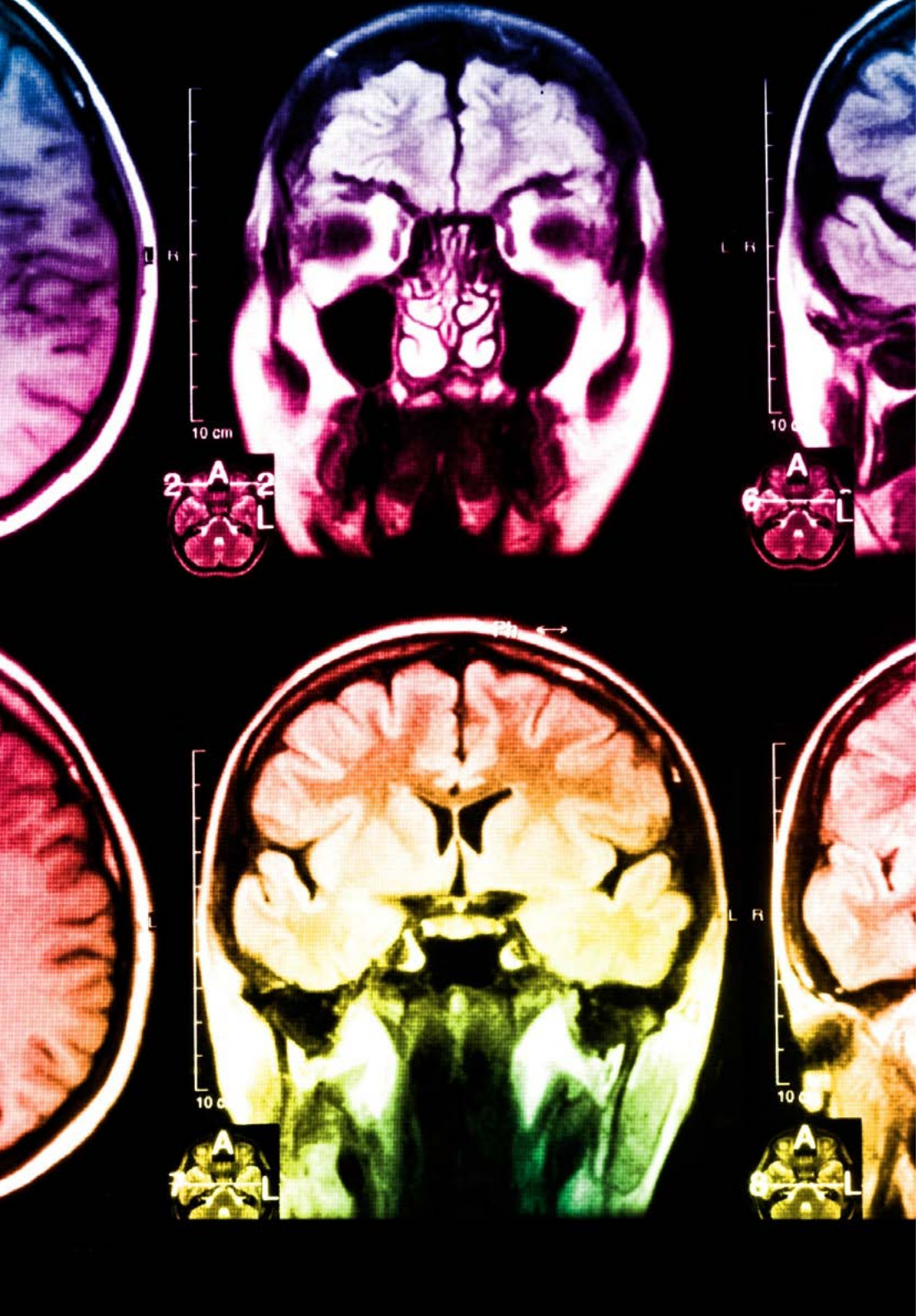
TECH coniuga efficacemente la metodologia del Caso di Studio con un sistema di apprendimento 100% online basato sulla ripetizione, che combina diversi elementi didattici in ogni lezione.

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:

Materiale 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.

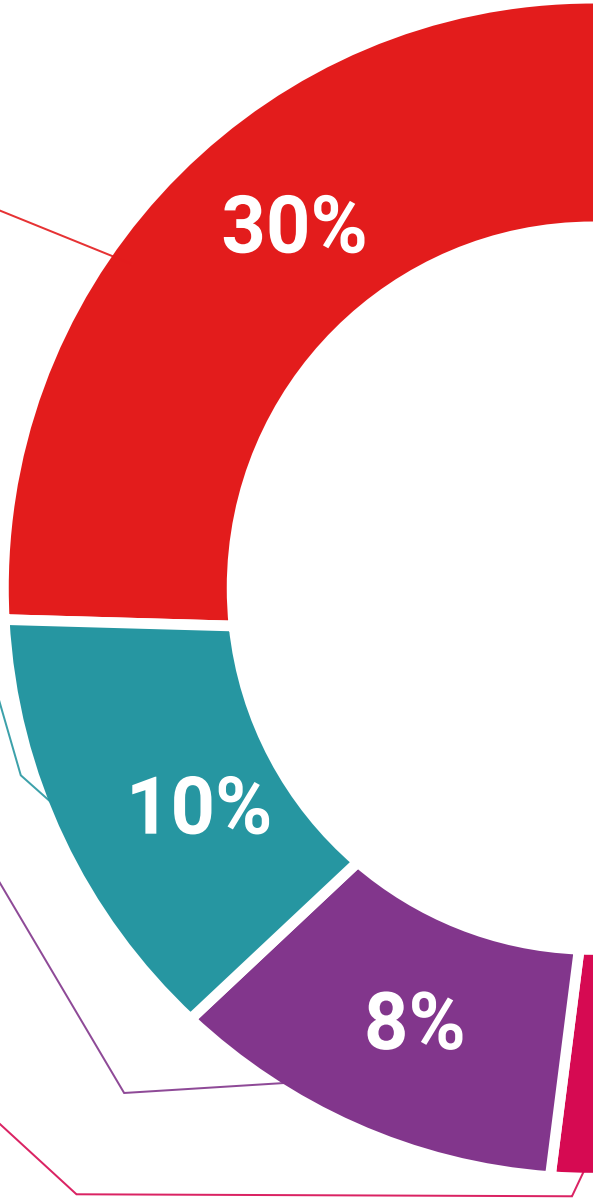
Imparare da un esperto rafforza la conoscenza e la memoria, costruisce la fiducia nelle nostre future decisioni difficili.

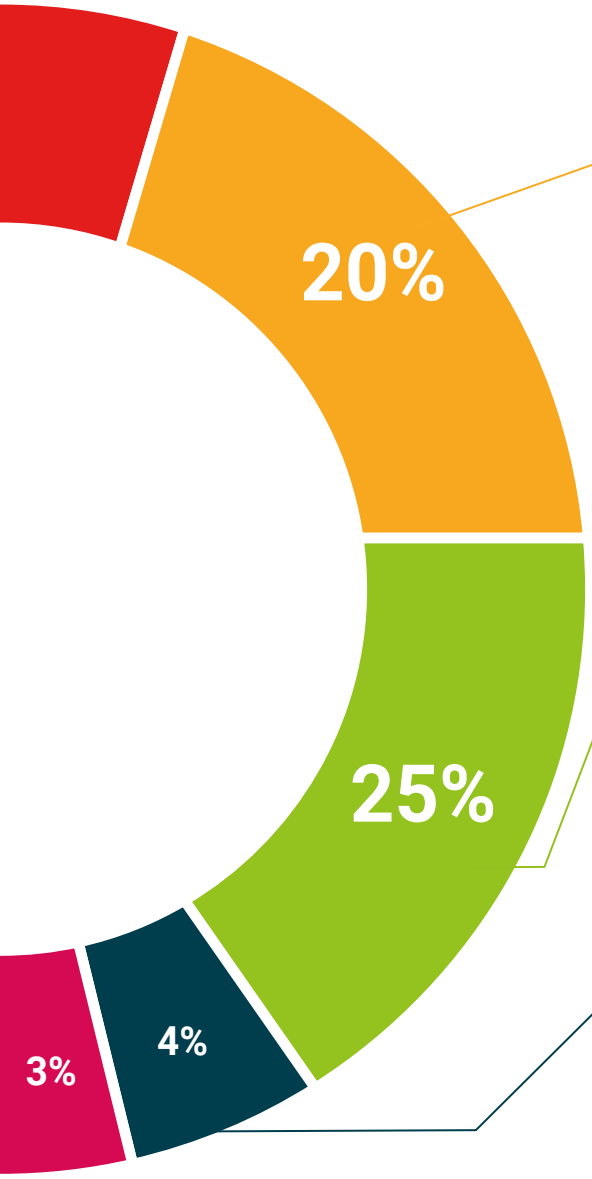
Pratiche di competenze e competenze

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.

09

Titolo

Il titolo di Master Semipresenziale in Sviluppo di Software garantisce, oltre alla specializzazione più rigorosa e aggiornata, l'accesso ad una qualifica di Master Semipresenziale rilasciata da TECH Università Tecnologica

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

Questo **Master Semipresenziale in Sviluppo di Software** possiede il programma più completo e aggiornato del panorama professionale e accademico.

Dopo aver superato le valutazioni, lo studente riceverà mediante lettera certificata, con ricevuta di ritorno, la corrispondente qualifica di Master Semipresenziale rilasciata da TECH Università Tecnologica, che accrediterà il superamento delle valutazioni e l'acquisizione delle competenze del programma.

Oltre alla qualifica, sarà possibile ottenere un certificato e un attestato dei contenuti delprogramma. A tal fine, sarà necessario contattare il proprio consulente accademico, che fornirà tutte le informazioni necessarie.

Titolo: **Master Semipresenziale in Sviluppo di Software**

Modalità: **Semipresenziale (Online + Tirocinio)**

Durata: **12 mesi**


tech università tecnologica

Conferisce il presente

DIPLOMA

a

Dott./Dott.ssa _____, con documento d'identità n°_____

Per aver completato con esito positivo e accreditato il programma di

MASTER SEMIPRESENZIALE

in

Sviluppo di Software

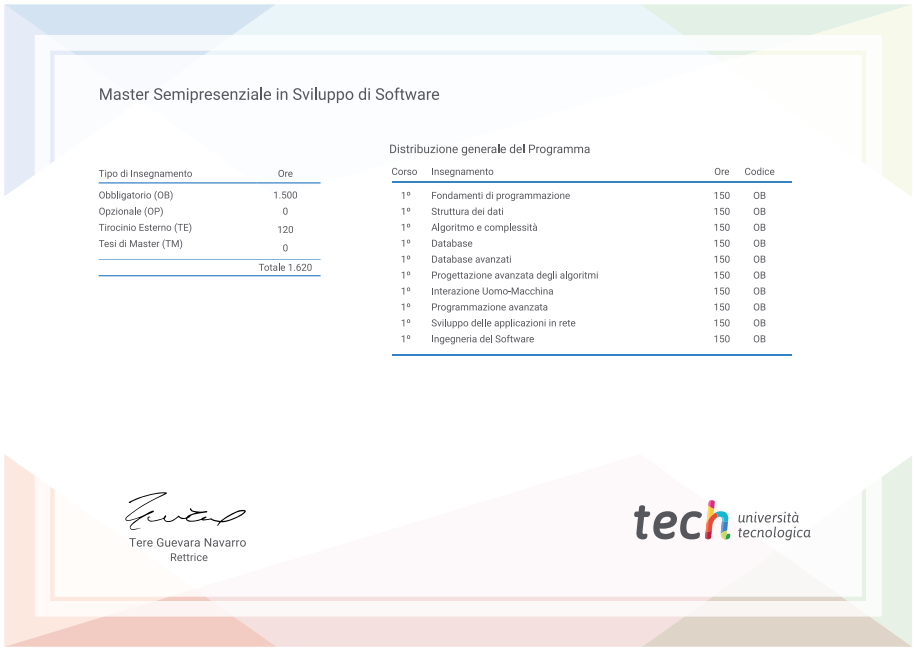
Si tratta di un titolo rilasciato da questa Università ed equivalente a 1.620 ore, con data di inizio gg/mm/aaaa e con data di fine gg/mm/aaaa.

TECH è un Istituto Privato di Istruzione Superiore riconosciuto dal Ministero della Pubblica Istruzione a partire dal 28 giugno 2018.

In data 17 Giugno 2020

Tere Guevara Navarro
Rettrice

codice unico TECH: AFWORD23S techtitute.com/titulos


Master Semipresenziale in Sviluppo di Software

Tipo di Insegnamento	Ore
Obbligatorio (OB)	1.500
Opzionale (OP)	0
Tirocinio Esterno (TE)	120
Tesi di Master (TM)	0
	Totale 1.620

Distribuzione generale del Programma

Corso	Insegnamento	Ore	Codice
1°	Fondamenti di programmazione	150	OB
1°	Struttura dei dati	150	OB
1°	Algoritmo e complessità	150	OB
1°	Database	150	OB
1°	Database avanzati	150	OB
1°	Progettazione avanzata degli algoritmi	150	OB
1°	Interazione Uomo-Macchina	150	OB
1°	Programmazione avanzata	150	OB
1°	Sviluppo delle applicazioni in rete	150	OB
1°	Ingegneria del Software	150	OB

Tere Guevara Navarro
Rettrice

tech università tecnologica

*Apostille dell'Aia. 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.

tech università tecnologica

Master Semipresenziale
Sviluppo di Software

Modalità: **Semipresenziale (Online + Tirocinio)**
Durata: **12 mesi**
Titolo: **TECH Università Tecnologica**

Master Semipresenziale
Sviluppo di Software

tech università tecnologica