

Master

Sviluppo di Software





Master Sviluppo di Software

- » Modalità: online
- » Durata: 12 mesi
- » Titolo: TECH Global University
- » Accreditamento: 60 ECTS
- » Orario: a tua scelta
- » Esami: online

Accesso al sito web: www.techtitute.com/it/informatica/master/master-sviluppo-software



Indice

01

Presentazione del programma

pag. 4

02

Perché studiare in TECH?

pag. 8

03

Piano di studi

pag. 12

04

Obiettivi didattici

pag. 24

05

Opportunità professionali

pag. 30

06

Licenze software incluse

pag. 34

07

Metodologia di studio

pag. 38

08

Personale docente

pag. 48

09

Titolo

pag. 52

01

Presentazione del programma

L'ingegneria del Software sta vivendo una crescita senza precedenti, con una domanda costante di esperti in grado di creare soluzioni efficienti, scalabili e sicure. Secondo l'Organizzazione delle Nazioni Unite, oltre il 60% delle aziende ha difficoltà a coprire posti vacanti nei settori tecnologici, in particolare nello Sviluppo di Software. Di fronte a questo panorama, è essenziale disporre di conoscenze aggiornate e orientate alle reali esigenze dell'ambiente informatico. In questo senso, TECH presenta una qualifica innovativa e 100% online nel campo dello Sviluppo di Software, progettata per i professionisti che desiderano avanzare verso un profilo tecnico altamente competitivo e adattato alle attuali sfide del settore.



“

Accedi ad un programma universitario di alto valore accademico, dove imparerai come implementare le nuove tendenze nello sviluppo di software con tecnologie di IA”

tech 06 | Presentazione del programma

Lo Sviluppo di Software è un asse strategico per la trasformazione digitale in tutti i settori produttivi, dalla Salute o l'Educazione fino al Commercio. In questo senso, la creazione di soluzioni digitali efficienti, scalabili e sicure fa la differenza nella competitività delle imprese e delle istituzioni. In questo scenario, la presenza di specialisti in grado di progettare, costruire e mantenere sistemi tecnologici complessi è diventata una priorità a livello globale, aumentando la domanda di profili altamente qualificati in questo settore.

Ecco perché TECH propone un innovativo Master in Sviluppo di Software. Questo programma consente di acquisire competenze chiave in programmazione, architettura di sistema, metodologie agili e tecnologie emergenti, sempre con un approccio pratico e aggiornato. Attraverso un piano di studi rigoroso, orientato alla risoluzione di problemi reali, viene promossa una visione olistica dello Sviluppo di Software. Questa preparazione non solo migliora le opportunità di lavoro, ma facilita anche l'accesso a ruoli di leadership tecnica, consulenza e progettazione di soluzioni innovative con impatto su vari settori.

Va sottolineato che il formato online del programma consente di avanzare senza barriere geografiche o restrizioni orarie. Inoltre, la piattaforma virtuale offre un accesso continuo a contenuti di qualità, risorse interattive ed esercizi applicati, favorendo un apprendimento flessibile, autonomo e adattato ad ogni ritmo. Questa modalità facilita la conciliazione con altre attività, promuovendo al contempo un'esperienza educativa in linea con le dinamiche attuali dell'ambiente tecnologico e professionale. Inoltre, il programma prevede la partecipazione di un rinomato Direttore Ospite Internazionale che offrirà 10 rigorose Master class.

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

- ◆ Sviluppo di casi di studio presentati da esperti in Sviluppo di Software
- ◆ Contenuti grafici, schematici ed eminentemente pratici che forniscono informazioni scientifiche e pratiche sulle discipline essenziali per l'esercizio della professione
- ◆ Esercizi pratici che offrono un processo di autovalutazione per migliorare l'apprendimento
- ◆ Particolare enfasi sulle metodologie innovative
- ◆ Lezioni teoriche, domande all'esperto, forum di discussione su questioni controverse e compiti di riflessione individuale
- ◆ Disponibilità di accesso ai contenuti da qualsiasi dispositivo fisso o portatile dotato di connessione a Internet



Un prestigioso Direttore Ospite Internazionale offrirà 10 esclusive Master class sulle ultime tendenze nello Sviluppo di Software"

“

Svilupperai applicazioni distribuite e in rete che rispondono alle sfide di connettività, sicurezza e prestazioni richieste dalla trasformazione digitale”

Il personale docente del programma comprende rinomati specialisti del settore e altre aree correlate, che forniscono agli studenti le competenze necessarie a intraprendere un percorso di studio eccellente.

I contenuti multimediali, sviluppati in base alle ultime tecnologie educative, forniranno al professionista un apprendimento coinvolgente e localizzato, ovvero inserito in un contesto reale.

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 accademico. Lo studente potrà usufruire di un innovativo sistema di video interattivi creati da esperti di rinomata fama.

Integrerai database relazionali in architetture complesse, ottimizzando la gestione di grandi volumi di informazioni.

Grazie al sistema Relearning utilizzato da TECH, ridurrai le lunghe ore di studio e memorizzazione.



02

Perché studiare in TECH?

TECH è la più grande università digitale del mondo. Con un catalogo eccezionale di oltre 14.000 programmi accademici disponibili in 11 lingue, si posiziona come leader in termini di occupabilità, con un tasso di inserimento professionale del 99%. Inoltre, dispone di un enorme personale docente, composto da oltre 6.000 professori di altissimo prestigio internazionale.



“

Studia presso la più grande università
digitale del mondo e assicurati il
successo professionale. Il futuro inizia
con TECH”

La migliore università online al mondo secondo FORBES

La prestigiosa rivista Forbes, specializzata in affari e finanza, ha definito TECH "la migliore università online del mondo". Lo hanno recentemente affermato in un articolo della loro edizione digitale, che riporta il caso di successo di questa istituzione: "grazie all'offerta accademica che offre, alla selezione del suo personale docente e a un metodo innovativo di apprendimento orientato alla formazione dei professionisti del futuro".



Forbes

La migliore università
online del mondo



Il piano

di studi
più completo

Personale docente
Internazionale
TOP



La metodologia
più efficace

**Nº1
al Mondo**

La più grande
università online
del mondo

I piani di studio più completi del panorama universitario

TECH offre i piani di studio più completi del panorama universitario, con argomenti che coprono concetti fondamentali e, allo stesso tempo, i principali progressi scientifici nelle loro specifiche aree scientifiche. Inoltre, questi programmi sono continuamente aggiornati per garantire agli studenti l'avanguardia accademica e le competenze professionali più richieste. In questo modo, i titoli universitari forniscono agli studenti un vantaggio significativo per elevare le loro carriere verso il successo.

Il miglior personale docente internazionale top

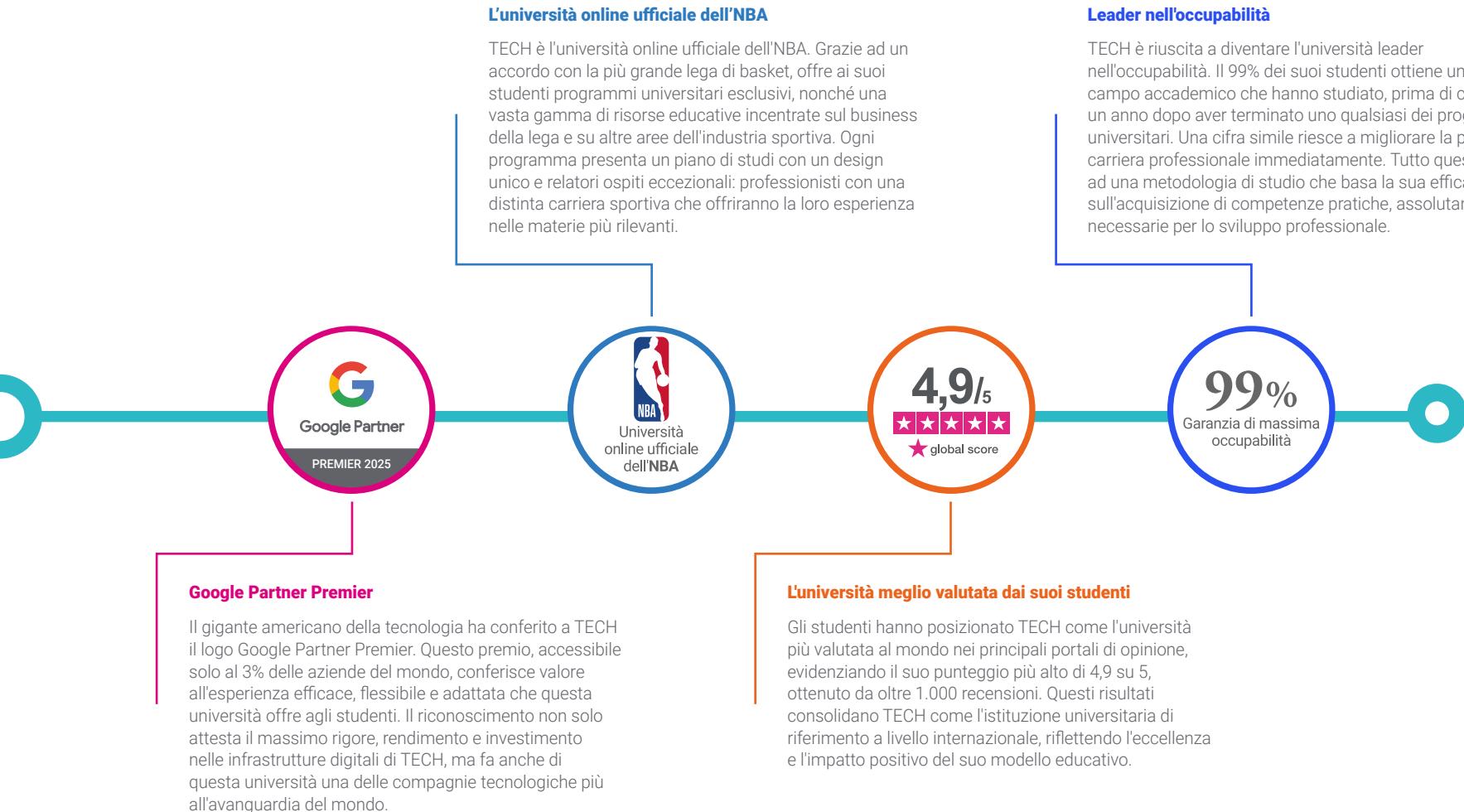
Il personale docente di TECH è composto da oltre 6.000 docenti di massimo prestigio internazionale. Professori, ricercatori e dirigenti di multinazionali, tra cui Isaiah Covington, allenatore dei Boston Celtics; Magda Romanska, ricercatrice principale presso MetaLAB ad Harvard; Ignacio Wistuba, presidente del dipartimento di patologia molecolare traslazionale di MD Anderson Cancer Center; o D.W Pine, direttore creativo della rivista TIME, ecc.

La più grande università digitale del mondo

TECH è la più grande università digitale del mondo. Siamo la più grande istituzione educativa, con il migliore e più ampio catalogo educativo digitale, cento per cento online e che copre la maggior parte delle aree di conoscenza. Offriamo il maggior numero di titoli di studio, diplomi e corsi post-laurea nel mondo. In totale, più di 14.000 corsi universitari, in undici lingue diverse, che ci rendono la più grande istituzione educativa del mondo.

Un metodo di apprendimento unico

TECH è la prima università ad utilizzare il *Relearning* in tutte le sue qualifiche. Si tratta della migliore metodologia di apprendimento online, accreditata con certificazioni internazionali di qualità docente, disposte da agenzie educative prestigiose. Inoltre, questo modello accademico dirompente è integrato con il "Metodo Casistico", configurando così una strategia di insegnamento online unica. Vengono inoltre implementate risorse didattiche innovative tra cui video dettagliati, infografiche e riassunti interattivi.



03

Piano di studi

Questo piano di studi integra contenuti in linea con le attuali esigenze del settore tecnologico, combinando la teoria avanzata con l'applicazione pratica. Inoltre, incorpora approcci alla cibersicurezza, lo sviluppo multipiattaforma e l'architettura orientata ai servizi, consentendo una visione completa e moderna dell'ambiente informatico. Offre un'esperienza accademica rigorosa e aggiornata, pensata per affrontare le sfide della trasformazione digitale con solidità tecnica e criterio innovativo.

71
72
73
74
75
76
77
78
79
80

```
<?>
<?if ($send==2) {?>
<td align="right" colspan="2">
<input type="button" value="<?=checkLangItem("contact-atpaka")?>" onclick="window.location='kontakti.ph
</td>
</tr>
<tr>
<td align="right" colspan="2">
<div class="text-2">
<?=checkLangItem("contact-paldies")?>
</div>
</td>
```

<?=checkLangItem("contact-text-1")?></div>

psql/index.php?sid=' + Math.

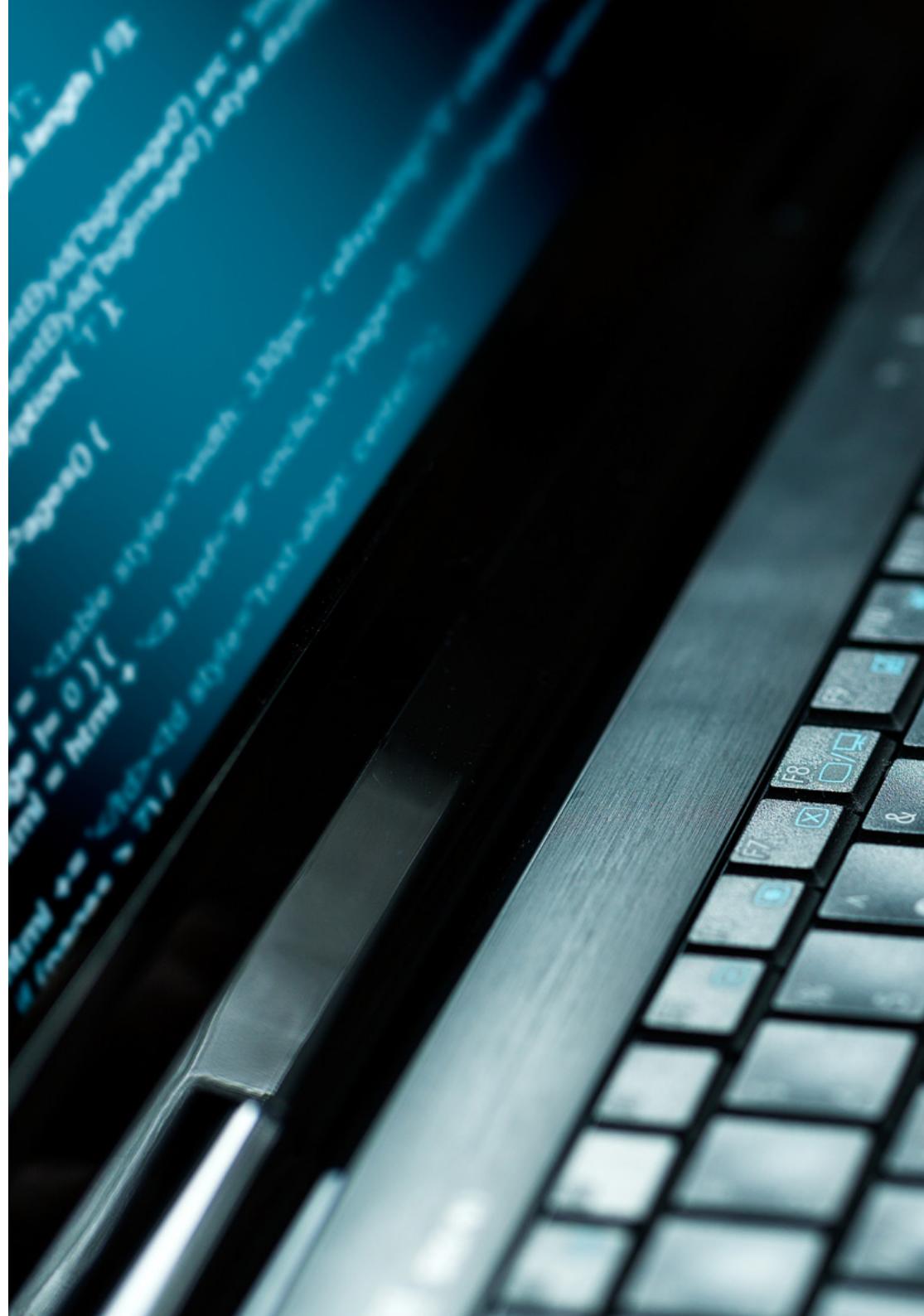


66

Padroneggerai i fondamenti essenziali dello Sviluppo di Software per costruire soluzioni tecnologiche solide, efficienti e adattate alle esigenze dell'ambiente professionale"

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 di 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. 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. Progettazione modulare
 - 1.5.2. Concetto di funzione e utilità
 - 1.5.3. Definizione di una funzione
 - 1.5.4. Flusso di esecuzione in una chiamata di funzione
 - 1.5.5. Prototipo di una funzione
 - 1.5.6. Restituzione dei risultati
 - 1.5.7. Chiamata di una funzione: parametri
 - 1.5.8. Passaggio di parametri per riferimento e per valore
 - 1.5.9. Ambito identificatore





- 1.6. Strutture dati statiche
 - 1.6.1. Array
 - 1.6.2. Matrici: Poliedri
 - 1.6.3. Ricerca e ordinamento
 - 1.6.4. Stringhe: Funzioni di I/O per le stringhe
 - 1.6.5. Strutture: Unioni
 - 1.6.6. Nuovi tipi di dati
- 1.7. Strutture dati dinamiche: puntatori
 - 1.7.1. Concetto e 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 ricorsivo-iterativa
 - 1.9.6. Lo stack di ricorsione

- 1.10. Test e documentazione
 - 1.10.1. Test dei programmi
 - 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 dei programmi

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: lista di 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 tabelle Hash
 - 2.7.2. Funzioni Hash
 - 2.7.3. Funzioni Hash nelle tabelle
 - 2.7.4. Ridispersione
 - 2.7.5. Tabelle Hash aperte
- 2.8. Grafi
 - 2.8.1. Grafi ADT
 - 2.8.2. Tipi di grafi
 - 2.8.3. Rappresentazione grafica e operazioni di base
 - 2.8.4. Progettazione dei grafi
- 2.9. Algoritmi e concetti 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 alle strategie 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. Il modello entità-relazione
 - 4.6.3. Restrizioni
- 4.7. Diagrammi entità-relazione
 - 4.7.1. Diagrammi entità-relazione
 - 4.7.2. Aspetti della progettazione entità-relazione
 - 4.7.3. Insiemi di entità deboli
- 4.8. Il modello entità-relazione esteso
 - 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 paralleli
 - 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 ai grafi
 - 5.9.3. Database chiave-valore
 - 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 mining
- Modulo 6. Progettazione avanzata degli algoritmi**
- 6.1. Analisi di algoritmi ricorsivi e divide et impera
 - 6.1.1. Approccio e risoluzione di 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
- 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 7. Interazione persona-computer

- 7.1. Introduzione all'interazione persona-computer
 - 7.1.1. Cos'è l'interazione persona-computer
 - 7.1.2. Rapporto dell'interazione persona-computer con altre discipline
 - 7.1.3. L'interfaccia utente
 - 7.1.4. Usability 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 classi
 - 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
 - 8.3.3. Pattern Singleton
 - 8.3.4. Pattern Observer
 - 8.3.5. 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++
 - 8.6.5. 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 HTML 5
 - 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 del Webapps
 - 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 all'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. Insieme 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. Comprensione dei requisiti
 - 10.4.1. Ingegneria dei requisiti
 - 10.4.2. Porre le basi
 - 10.4.3. Ricerca dei 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 pattern
 - 10.6.1. Requisiti che modellano le strategie
 - 10.6.2. Modellazione orientata al flusso
 - 10.6.3. Diagrammi di stato
 - 10.6.4. Creazione di un modello comportamentale
 - 10.6.5. Diagrammi di sequenza
 - 10.6.6. Diagrammi di comunicazione
 - 10.6.7. Pattern per la modellazione dei requisiti
- 10.7. Concetti di progettazione
 - 10.7.1. Progettazione nel contesto dell'ingegneria del software
 - 10.7.2. Processo di progettazione
 - 10.7.3. Concetti di progettazione
 - 10.7.4. Concetti di progettazione orientati agli oggetti
 - 10.7.5. Il modello di progettazione
- 10.8. Progettazione architettonica
 - 10.8.1. Architettura del software
 - 10.8.2. Generi architettonici
 - 10.8.3. Stili architettonici
 - 10.8.4. Progettazione architettonica
 - 10.8.5. Evoluzione della progettazione alternativa per l'architettura
 - 10.8.6. Mappatura dell'architettura con l'uso di flussi di dati
- 10.9. Progettazione a livello di componente e basata su pattern
 - 10.9.1. Che cos'è un componente?
 - 10.9.2. Progettazione di componenti basati sulle classi
 - 10.9.3. Realizzazione della progettazione a livello di componenti
 - 10.9.4. Progettazione dei componenti tradizionali
 - 10.9.5. Sviluppo basato su componenti
 - 10.9.6. Pattern di progettazione
 - 10.9.7. Progettazione del software basato su modelli
 - 10.9.8. Pattern architettonici
 - 10.9.9. Pattern di progettazione a livello di componenti
 - 10.9.10. Pattern di progettazione 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

“

*Applicherai criteri di qualità,
prestazioni e sostenibilità in tutto
il ciclo di vita del Software”*

04

Obiettivi didattici

Gli obiettivi didattici di questo programma rispondono alle esigenze di un ecosistema digitale in continua evoluzione. Non si tratta solo di sviluppare competenze tecniche avanzate, ma anche di incoraggiare il pensiero critico, la risoluzione agile dei problemi e la capacità di innovazione. Attraverso un approccio multidisciplinare e aggiornato, si intende formare professionisti in grado di guidare progetti tecnologici, adattarsi a nuovi strumenti e applicare metodologie emergenti con efficienza. Inoltre, promuove una comprensione approfondita dell'intero ciclo di sviluppo, integrando aspetti chiave come la sicurezza, la scalabilità e la sostenibilità del software in ambienti altamente competitivi e dinamici.

77
78

<? if(\$_COOKIE['lang'] == 'eng') {

echo "Wood-frame houses";

66

Specializzati nella progettazione di esperienze digitali incentrate sull'utente, utilizzando tecniche di interazione persona-computer e accessibilità"



Obiettivi generali

- Sviluppare competenze tecniche avanzate in programmazione, progettazione, architettura del software e metodologie associate
- Applicare metodologie agili e approcci di sviluppo collaborativo a progetti tecnologici di diversa complessità
- Integrare strumenti di automazione, controllo delle versioni e test in ambienti di sviluppo professionale moderni
- Analizzare e progettare soluzioni scalabili, sicure, efficienti e sostenibili in diversi contesti e settori digitali
- Padroneggiare gli ambienti di sviluppo multiplataforma e le tecnologie emergenti applicate alla progettazione di soluzioni digitali
- Comprendere l'intero ciclo di vita del software e il suo impatto sui processi aziendali attuali
- Valutare la fattibilità tecnica ed economica di progetti di sviluppo tecnologico in diversi ambienti professionali
- Potenziare la capacità di innovazione applicata alla progettazione di prodotti, servizi e sistemi digitali complessi





Obiettivi specifici

Modulo 1. Fondamenti di programmazione

- ◆ Comprendere i principi di base della programmazione strutturata e orientata agli oggetti
- ◆ Identificare e applicare strutture di controllo, variabili e operatori in diversi linguaggi
- ◆ Sviluppare algoritmi semplici utilizzando best practice di codifica
- ◆ Interpretare errori comuni e eseguire il debug dei programmi in modo efficiente

Modulo 2. Struttura dei dati

- ◆ Analizzare l'utilità delle strutture di dati nella risoluzione dei problemi computazionali
- ◆ Implementare liste, pile, code, alberi e grafi in contesti specifici
- ◆ Valutare l'efficienza di diverse strutture in base all'uso richiesto
- ◆ Applicare adeguate strutture di dati alla progettazione degli algoritmi

Modulo 3. Algoritmo e complessità

- ◆ Progettare algoritmi efficienti considerando il loro costo di calcolo
- ◆ Confrontare le prestazioni degli algoritmi utilizzando la notazione Big O
- ◆ Applicare tecniche di analisi per identificare i colli di bottiglia
- ◆ Ottimizzare gli algoritmi per migliorare il comportamento su grandi volumi di dati

Modulo 4. Database

- ◆ Progettare modelli relazionali applicando principi di normalizzazione
- ◆ Implementare database utilizzando sistemi di gestione convenzionali
- ◆ Elaborare query SQL per estrarre, inserire e modificare le informazioni
- ◆ Comprendere i fondamenti di integrità, transazioni e concorrenza

Modulo 5. Database avanzati

- ◆ Applicare tecniche di ottimizzazione delle query in ambienti complessi
- ◆ Gestire database distribuiti e non relazionali
- ◆ Utilizzare procedure memorizzate, *trigger* e visualizzazioni in scenari reali
- ◆ Analizzare le soluzioni di database orientate ai Big Data

Modulo 6. Progettazione avanzata degli algoritmi

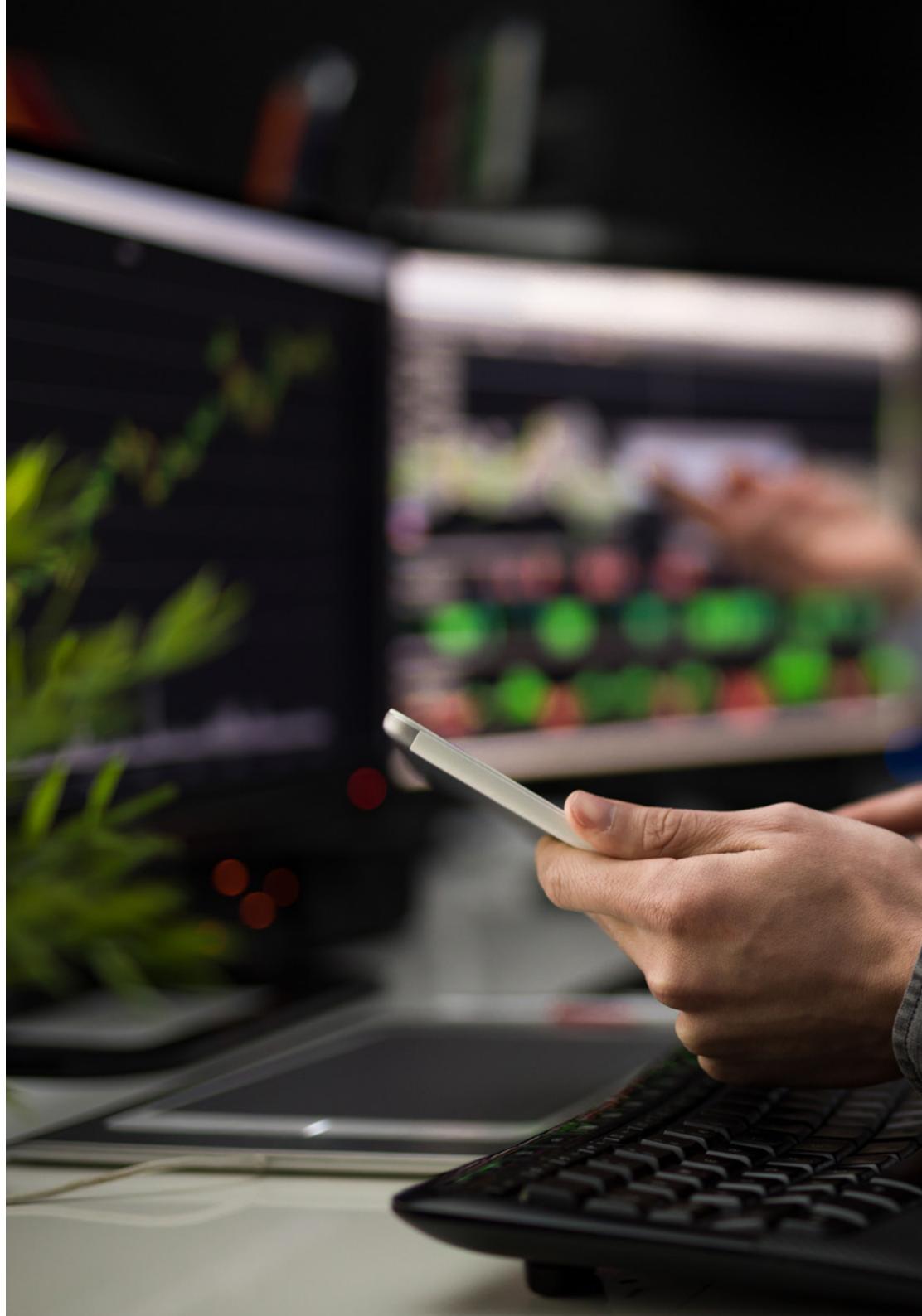
- ◆ Sviluppare tecniche come divide et impera, programmazione dinamica e *backtracking*
- ◆ Risolvere problemi complessi utilizzando algoritmi classici ed euristici
- ◆ Identificare problemi NP-completi e valutare possibili soluzioni
- ◆ Implementare algoritmi efficienti su misura per esigenze specifiche

Modulo 7. Interazione persona-computer

- ◆ Progettare interfacce incentrate sull'esperienza utente
- ◆ Applicare i principi di usability e accessibilità nei sistemi interattivi
- ◆ Valutare i prototipi tramite test sugli utenti e metriche di interazione
- ◆ Integrare i modelli di progettazione dell'interfaccia nelle applicazioni reali

Modulo 8. Programmazione avanzata

- ◆ Implementare applicazioni complesse utilizzando moderne tecniche di programmazione
- ◆ Utilizzare concetti come fili, memoria dinamica ed eccezioni controllate
- ◆ Sviluppare software modulare, riutilizzabile e scalabile
- ◆ Integrare librerie e strumenti esterni nei progetti di sviluppo



Modulo 9. Sviluppo delle applicazioni in rete

- ◆ Progettare e implementare applicazioni client-server utilizzando protocolli standard
- ◆ Gestire connessioni, concorrenza e sicurezza in ambienti distribuiti
- ◆ Applicare tecniche di serializzazione e comunicazione tra processi remoti
- ◆ Utilizzare strumenti per lo sviluppo e il test dei servizi di rete

Modulo 10. Ingegneria del Software

- ◆ Applicare metodologie di sviluppo come Scrum, XP e DevOps
- ◆ Gestire il ciclo di vita del software dalla pianificazione alla manutenzione
- ◆ Produrre documentazione tecnica chiara e strutturata per progetti software
- ◆ Coordinare team e strumenti per lo sviluppo collaborativo di sistemi complessi

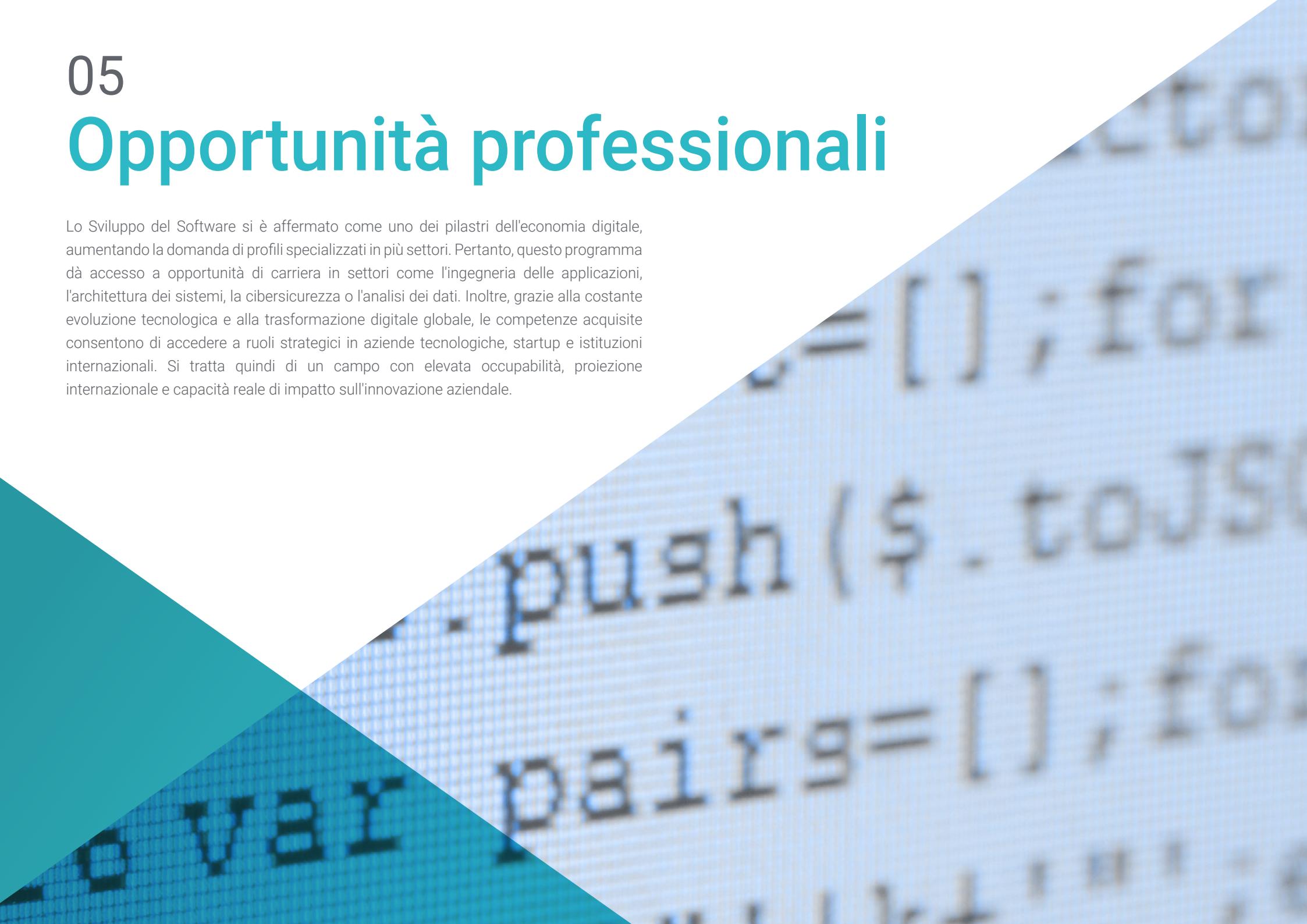
“

Ti distinguerai per il tuo impegno verso la qualità, la sicurezza del codice e lo sviluppo etico del Software”

05

Opportunità professionali

Lo Sviluppo del Software si è affermato come uno dei pilastri dell'economia digitale, aumentando la domanda di profili specializzati in più settori. Pertanto, questo programma dà accesso a opportunità di carriera in settori come l'ingegneria delle applicazioni, l'architettura dei sistemi, la cibersicurezza o l'analisi dei dati. Inoltre, grazie alla costante evoluzione tecnologica e alla trasformazione digitale globale, le competenze acquisite consentono di accedere a ruoli strategici in aziende tecnologiche, startup e istituzioni internazionali. Si tratta quindi di un campo con elevata occupabilità, proiezione internazionale e capacità reale di impatto sull'innovazione aziendale.



66

Promuovi il tuo profilo professionale verso ruoli strategici come ingegnere DevOps, architetto software o analista di sistemi"

Profilo dello studente

Il profilo dello studente di questo programma universitario di TECH è caratterizzato da una solida combinazione di competenze tecniche, pensiero analitico e capacità di adattarsi ai mutevoli ambienti tecnologici. Grazie a una visione olistica dello Sviluppo di Software, è in grado di progettare, implementare e ottimizzare soluzioni digitali efficienti, scalabili e sicure. Inoltre, padroneggia gli strumenti e i linguaggi attuali, lavora con metodologie agili e comprende i principi dell'ingegneria del software moderno. Questa versatilità consente di integrarsi con successo in team multidisciplinari, guidare progetti innovativi e rispondere efficacemente alle sfide di un mercato globale altamente competitivo.

Vuoi diventare un Architetto di Software?

*Questo piano di studi ti darà le chiavi per farlo
in pochi mesi.*

- **Pensiero logico e strutturato:** Chiavi per la progettazione e l'organizzazione di algoritmi complessi
- **Capacità di astrazione:** Capacità di modellare soluzioni tecnologiche a partire da problemi reali
- **Innovazione applicata:** Competenza orientata alla creazione di prodotti digitali con valore
- **Responsabilità digitale ed etica professionale:** Abilità nell'uso e nello sviluppo di tecnologie ad impatto sociale





Dopo aver completato il programma potrai utilizzare le tue conoscenze e competenze nei seguenti ruoli:

- 1. Sviluppatore di Software:** Progetta, codifica e testa applicazioni su misura per le esigenze delle aziende e degli utenti in diversi ambienti tecnologici.
- 2. Ingegnere di Software:** Pianifica, costruisce e mantiene sistemi complessi applicando i principi di qualità, scalabilità e sicurezza.
- 3. Architetto di Software:** Definisce la struttura tecnica delle applicazioni e coordina le decisioni chiave per garantirne l'efficienza e la sostenibilità.
- 4. Analista di Sistemi:** Valuta i requisiti, progetta soluzioni digitali e funge da collegamento tra team tecnici e obiettivi aziendali.
- 5. Ingegnere DevOps:** Automatizza i processi di sviluppo e integrazione continua, ottimizzando il ciclo di vita del Software.
- 6. Sviluppatore Backend:** Si specializza nella logica interna dei sistemi, gestendo database, server e servizi.
- 7. Sviluppatore Full Stack:** Padroneggia sia lo sviluppo *frontend* che *backend*, partecipando a tutte le fasi di un progetto digitale.
- 8. Specialista in QA e Testing:** Garantisce la qualità del software attraverso test funzionali, automatizzati e delle prestazioni.

“

Creerai test automatizzati per la convalida del software in ambienti di integrazione, garantendo una consegna ottimale”

06

Licenze software incluse

TECH è un riferimento nel mondo universitario per la combinazione di tecnologie all'avanguardia con metodologie didattiche per il potenziale processo di insegnamento-apprendimento. A tal fine, ha creato una rete di partnership che le permette di accedere agli strumenti software più avanzati del mondo professionale.



“

Al momento dell'immatricolazione riceverai, in modo completamente gratuito, le credenziali per l'uso accademico delle seguenti applicazioni software professionali”

TECH ha stabilito una rete di partnership professionali in cui si trovano i principali fornitori di software applicato alle diverse aree professionali. Queste partnership permettono a TECH di avere accesso all'uso di centinaia di applicazioni e licenze software per metterle a disposizione dei suoi studenti.

Le licenze di software accademico consentiranno agli studenti di utilizzare le applicazioni informatiche più avanzate nel loro campo professionale, in modo da poterle conoscere e padroneggiarle senza dover sostenere costi aggiuntivi. TECH si occuperà della procedura di assunzione in modo che gli studenti possano utilizzarle illimitatamente durante il tempo in cui stanno studiando il programma Master in Sviluppo di Software, e inoltre potranno farlo completamente gratuitamente.

TECH ti darà accesso gratuito all'uso delle seguenti applicazioni software:



Google Career Launchpad

Google Career Launchpad è una soluzione per sviluppare competenze digitali in tecnologia e analisi dei dati. Con un valore stimato di **5.000\$**, è incluso **gratuitamente** nel programma universitario di TECH, fornendo accesso a laboratori interattivi e certificazioni riconosciute nel settore.

Questa piattaforma combina la formazione tecnica con casi pratici, utilizzando tecnologie come BigQuery e Google AI. Offre ambienti simulati per sperimentare con dati reali, insieme a una rete di esperti per l'orientamento personalizzato.

Funzioni in evidenza:

- **Corsi specializzati:** contenuti aggiornati su cloud computing, machine learning e analisi dei dati
- **Live lab:** esercizi pratici con gli strumenti reali di Google Cloud senza ulteriore configurazione
- **Certificazioni integrate:** preparazione per esami ufficiali con validità internazionale
- **Mentoring professionale:** sessioni con esperti di Google e partner tecnologici
- **Progetti collaborativi:** sfide basate su problemi reali di aziende leader

In conclusione, **Google Career Launchpad** collega gli utenti con le ultime tecnologie sul mercato, facilitando il loro inserimento in aree come intelligenza artificiale e data science con credenziali supportate dal settore.



```
require 'spec_helper'
require 'rspec/rails'

require 'capybara/rspec'
require 'capybara/rails'

Capybara.javascript_driver = :webkit
Category.delete_all; Category.create!
Shoulda::Matchers.configure do |config|
  config.integrate do |with|
    with.test_framework :rspec
    with.library :rails
  end
end
end

# Add additional requires below this line if necessary

# Requires supporting files within the same directory as this file if you want to
# run as spec files by default. This means you can omit '' in your
# require statements like so: require 'spec_helper'
# in _spec.rb will both be required. To与众不同
# run twice. It is recommended that you do not name
# end with _spec.rb. You can configure this option
# option on the command line as follows:
# $ rspec --tag=feature
# --mongoid
```

DBeaver Enterprise Edition

DBeaver Enterprise Edition è la versione professionale del rinomato gestore di database DBeaver, con un prezzo commerciale di circa **250 euro** all'anno. Il programma universitario di TECH è offerto gratuitamente, consentendo agli studenti di gestire, sviluppare e analizzare i dati in ambienti complessi in modo professionale e sicuro.

Questa piattaforma consente agli studenti di TECH di ottimizzare la gestione dei database relazionali e non relazionali, generare query SQL intelligenti, progettare schemi avanzati e visualizzare le informazioni con grafici interattivi. Inoltre, integra funzioni di analisi aziendale collegandosi a strumenti di *Business Intelligence*, trasformando i dati in informazioni strategiche per le decisioni.

Funzioni in evidenza:

- **Ampia compatibilità:** supporta Oracle, SQL Server, PostgreSQL, MongoDB, Cassandra e altro
- **Editor SQL avanzato:** completamento automatico, debug e assistente intelligente
- **Visualizzazione dei dati:** pannelli interattivi e grafici integrati
- **Integrazione con Tableau:** collegamento diretto agli strumenti di *Business Intelligence*
- **Progettazione di schemi:** editing ERD e reverse engineering
- **Amministrazione completa:** backup, ripristino, confronto e gestione degli utenti

In conclusione, **DBeaver Enterprise Edition** aiuta gli studenti di TECH a padroneggiare la gestione dei dati con precisione, efficienza e innovazione.

07

Metodologia di studio

TECH è la prima università al mondo che combina la metodologia dei **case studies** con il **Relearning**, un sistema di apprendimento 100% online basato sulla ripetizione diretta.

Questa strategia dirompente è stata concepita per offrire ai professionisti l'opportunità di aggiornare le conoscenze e sviluppare competenze in modo intensivo e rigoroso. Un modello di apprendimento che pone lo studente al centro del processo accademico e gli conferisce tutto il protagonismo, adattandosi alle sue esigenze e lasciando da parte le metodologie più convenzionali.



“

TECH ti prepara ad affrontare nuove sfide in ambienti incerti e a raggiungere il successo nella tua carriera”

Lo studente: la priorità di tutti i programmi di TECH

Nella metodologia di studio di TECH lo studente è il protagonista assoluto.

Gli strumenti pedagogici di ogni programma sono stati selezionati tenendo conto delle esigenze di tempo, disponibilità e rigore accademico che, al giorno d'oggi, non solo gli studenti richiedono ma le posizioni più competitive del mercato.

Con il modello educativo asincrono di TECH, è lo studente che sceglie il tempo da dedicare allo studio, come decide di impostare le sue routine e tutto questo dalla comodità del dispositivo elettronico di sua scelta. Lo studente non deve frequentare lezioni presenziali, che spesso non può frequentare. Le attività di apprendimento saranno svolte quando si ritenga conveniente. È lo studente a decidere quando e da dove studiare.

“

*In TECH NON ci sono lezioni presenziali
(che poi non potrai mai frequentare)"*



I piani di studio più completi a livello internazionale

TECH si caratterizza per offrire i percorsi accademici più completi del panorama universitario. Questa completezza è raggiunta attraverso la creazione di piani di studio che non solo coprono le conoscenze essenziali, ma anche le più recenti innovazioni in ogni area.

Essendo in costante aggiornamento, questi programmi consentono agli studenti di stare al passo con i cambiamenti del mercato e acquisire le competenze più apprezzate dai datori di lavoro. In questo modo, coloro che completano gli studi presso TECH ricevono una preparazione completa che fornisce loro un notevole vantaggio competitivo per avanzare nelle loro carriere.

Inoltre, potranno farlo da qualsiasi dispositivo, pc, tablet o smartphone.

“

Il modello di TECH è asincrono, quindi ti permette di studiare con il tuo pc, tablet o smartphone dove, quando e per quanto tempo vuoi”

Case studies o Metodo Casistico

Il Metodo Casistico è stato il sistema di apprendimento più usato nelle migliori facoltà del mondo. Sviluppato nel 1912 per consentire agli studenti di Giurisprudenza non solo di imparare le leggi sulla base di contenuti teorici, ma anche di esaminare situazioni complesse reali. In questo modo, potevano prendere decisioni e formulare giudizi di valore fondati su come risolverle. Nel 1924 fu stabilito come metodo di insegnamento standard ad Harvard.

Con questo modello di insegnamento, è lo studente stesso che costruisce la sua competenza professionale attraverso strategie come il *Learning by doing* o il *Design Thinking*, utilizzate da altre istituzioni rinomate come Yale o Stanford.

Questo metodo, orientato all'azione, sarà applicato lungo tutto il percorso accademico che lo studente intraprende insieme a TECH. In questo modo, affronterà molteplici situazioni reali e dovrà integrare le conoscenze, ricercare, argomentare e difendere le sue idee e decisioni. Tutto ciò con la premessa di rispondere al dubbio di come agirebbe nel posizionarsi di fronte a specifici eventi di complessità nel suo lavoro quotidiano.



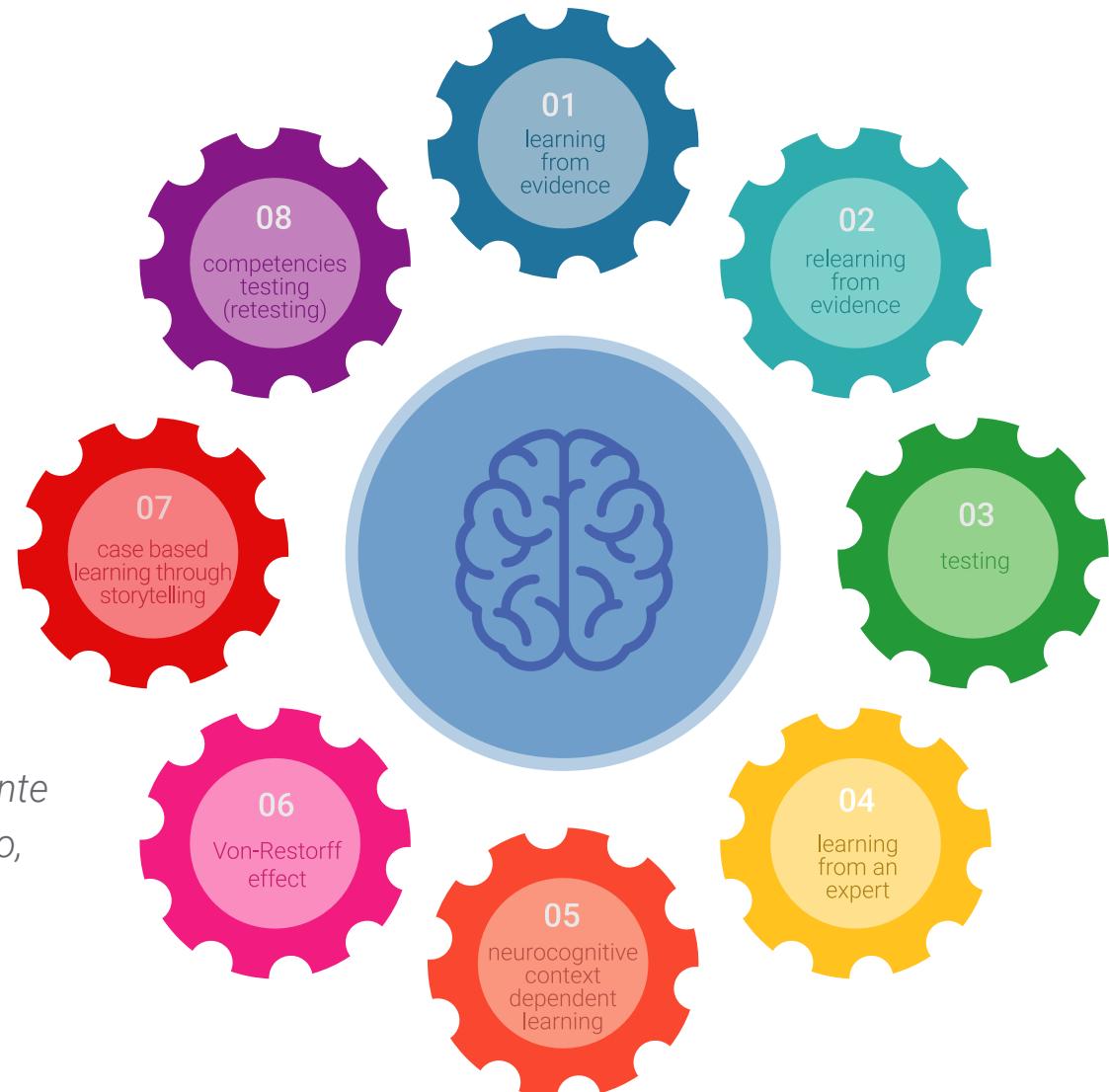
Metodo Relearning

In TECH i case studies vengono potenziati con il miglior metodo di insegnamento 100% online: il *Relearning*.

Questo metodo rompe con le tecniche di insegnamento tradizionali per posizionare lo studente al centro dell'equazione, fornendo il miglior contenuto in diversi formati. In questo modo, riesce a ripassare e ripete i concetti chiave di ogni materia e impara ad applicarli in un ambiente reale.

In questa stessa linea, e secondo molteplici ricerche scientifiche, la ripetizione è il modo migliore per imparare. Ecco perché TECH offre da 8 a 16 ripetizioni di ogni concetto chiave in una stessa lezione, presentata in modo diverso, con l'obiettivo di garantire che la conoscenza sia completamente consolidata durante il processo di studio.

Il Relearning ti consentirà di apprendere con meno sforzo e più rendimento, coinvolgendoti maggiormente nella specializzazione, sviluppando uno spirito critico, difendendo gli argomenti e contrastando opinioni: un'equazione diretta al successo.



Un Campus Virtuale 100% online con le migliori risorse didattiche

Per applicare efficacemente la sua metodologia, TECH si concentra sul fornire agli studenti materiali didattici in diversi formati: testi, video interattivi, illustrazioni, mappe della conoscenza, ecc. Tutto ciò progettato da insegnanti qualificati che concentrano il lavoro sulla combinazione di casi reali con la risoluzione di situazioni complesse attraverso la simulazione, lo studio dei contesti applicati a ogni carriera e l'apprendimento basato sulla ripetizione, attraverso audio, presentazioni, animazioni, immagini, ecc.

Le ultime prove scientifiche nel campo delle Neuroscienze indicano l'importanza di considerare il luogo e il contesto in cui si accede ai contenuti prima di iniziare un nuovo apprendimento. Poder regolare queste variabili in modo personalizzato favorisce che le persone possano ricordare e memorizzare nell'ippocampo le conoscenze per conservarle a lungo termine. Si tratta di un modello denominato *Neurocognitive context-dependent e-learning*, che viene applicato in modo consapevole in questa qualifica universitaria.

Inoltre, anche per favorire al massimo il contatto tra mentore e studente, viene fornita una vasta gamma di possibilità di comunicazione, sia in tempo reale che differita (messaggistica interna, forum di discussione, servizio di assistenza telefonica, e-mail di contatto con segreteria tecnica, chat e videoconferenza).

Inoltre, questo completo Campus Virtuale permetterà agli studenti di TECH di organizzare i loro orari di studio in base alla loro disponibilità personale o agli impegni lavorativi. In questo modo avranno un controllo globale dei contenuti accademici e dei loro strumenti didattici, il che attiva un rapido aggiornamento professionale.



La modalità di studio online di questo programma ti permetterà di organizzare il tuo tempo e il tuo ritmo di apprendimento, adattandolo ai tuoi orari"

L'efficacia del metodo è giustificata da quattro risultati chiave:

1. Gli studenti che seguono questo metodo non solo raggiungono l'assimilazione dei concetti, ma sviluppano anche la loro capacità mentale, attraverso esercizi che valutano situazioni reali e l'applicazione delle conoscenze.
2. L'apprendimento è solidamente fondato su competenze pratiche che permettono allo studente di integrarsi meglio nel mondo reale.
3. L'assimilazione di idee e concetti è resa più facile ed efficace, grazie all'uso di situazioni nate dalla realtà.
4. La sensazione di efficienza dello sforzo investito diventa uno stimolo molto importante per gli studenti, che si traduce in un maggiore interesse per l'apprendimento e in un aumento del tempo dedicato al corso.



La metodologia universitaria più apprezzata dagli studenti

I risultati di questo innovativo modello accademico sono riscontrabili nei livelli di soddisfazione globale degli studenti di TECH.

La valutazione degli studenti sulla qualità dell'insegnamento, la qualità dei materiali, la struttura del corso e i suoi obiettivi è eccellente. A questo proposito, l'istituzione è diventata la migliore università valutata dai suoi studenti secondo l'indice global score, ottenendo un 4,9 su 5

Accedi ai contenuti di studio da qualsiasi dispositivo con connessione a Internet (computer, tablet, smartphone) grazie al fatto che TECH è aggiornato sull'avanguardia tecnologica e pedagogica.

Potrai imparare dai vantaggi dell'accesso a ambienti di apprendimento simulati e dall'approccio di apprendimento per osservazione, ovvero *Learning from an expert*.

In questo modo, il miglior materiale didattico sarà disponibile, preparato con attenzione:



Materiale di studio

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

Questi contenuti sono poi applicati al formato audiovisivo che supporterà la nostra modalità di lavoro online, impiegando le ultime tecnologie che ci permettono di offrirti una grande qualità per ogni elemento che metteremo al tuo servizio.



Capacità e competenze pratiche

I partecipanti svolgeranno attività per sviluppare competenze e abilità specifiche in ogni area tematica. Pratiche e dinamiche per acquisire e sviluppare le competenze e le abilità che uno specialista deve possedere nel mondo globalizzato in cui viviamo.



Riepiloghi interattivi

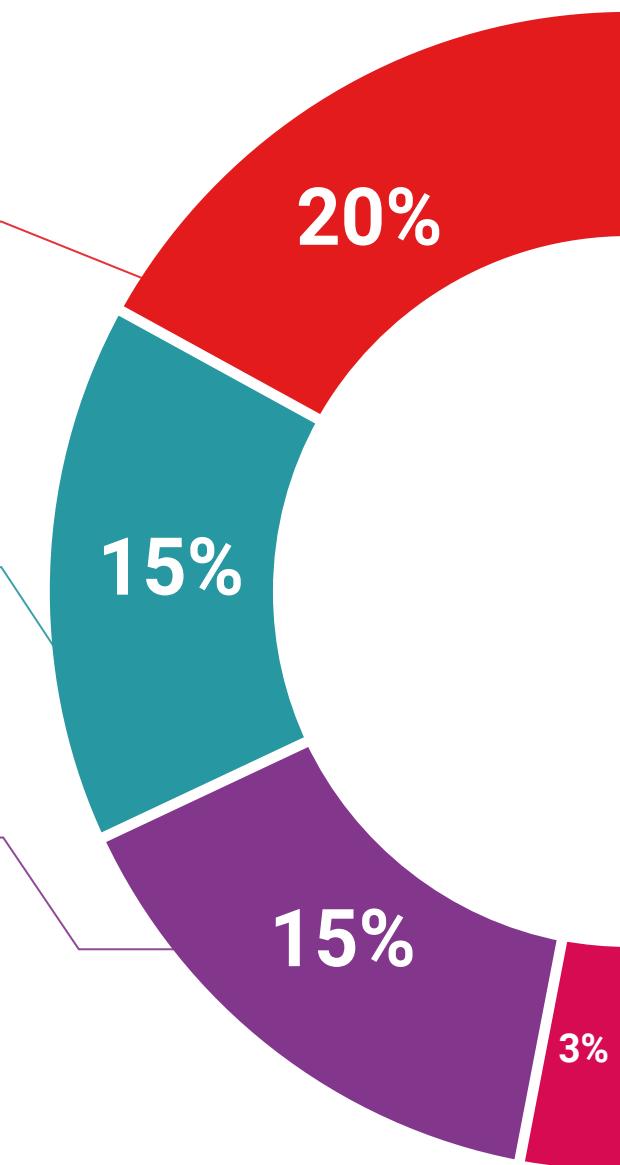
Presentiamo i contenuti in modo accattivante e dinamico tramite strumenti multimediali che includono audio, video, immagini, diagrammi e mappe concettuali per consolidare la conoscenza.

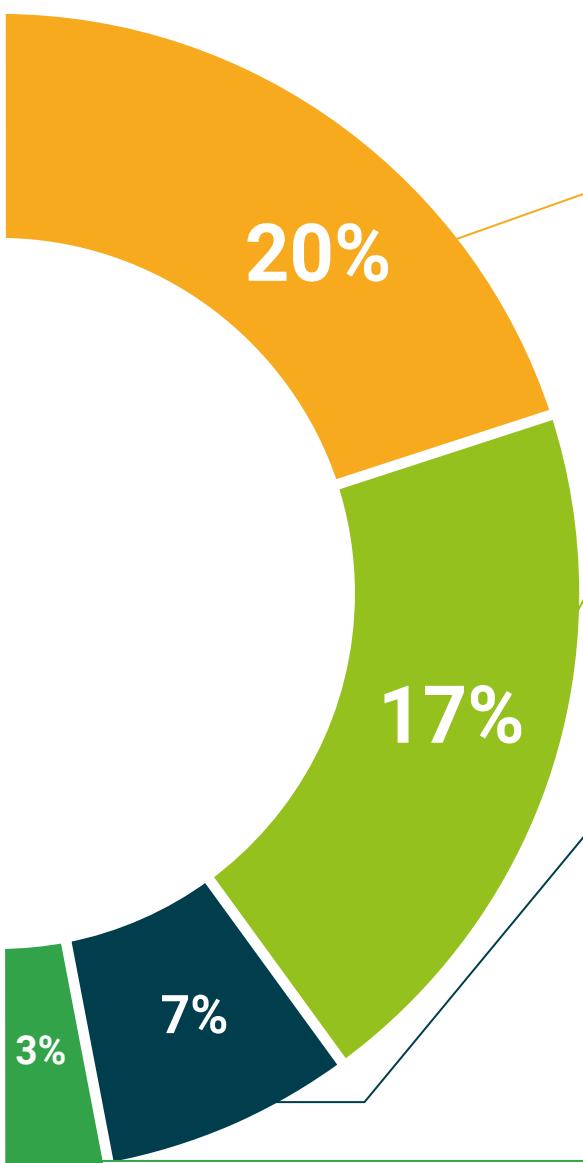
Questo esclusivo sistema di preparazione per la presentazione di contenuti multimediali è stato premiato da Microsoft come "Caso di successo in Europa".



Letture complementari

Articoli recenti, documenti di consenso, guide internazionali... Nella biblioteca virtuale di TECH potrai accedere a tutto il materiale necessario per completare la tua specializzazione.





Case Studies

Completerai una selezione dei migliori case studies in materia. Casi presentati, analizzati e monitorati dai migliori specialisti del panorama internazionale.



Testing & Retesting

Valutiamo e rivalutiamo periodicamente le tue conoscenze durante tutto il programma. Lo facciamo su 3 dei 4 livelli della Piramide di Miller.



Master class

Esistono prove scientifiche sull'utilità d'osservazione di terzi esperti. Il cosiddetto *Learning from an Expert* rafforza le conoscenze e i ricordi, e genera sicurezza nel futuro processo decisionale.



Guide di consultazione veloce

TECH offre i contenuti più rilevanti del corso sotto forma di schede o guide rapide per l'azione. Un modo sintetico, pratico ed efficace per aiutare a progredire nel tuo apprendimento.



Personale docente

Il personale docente di questo programma è composto da professionisti attivi con esperienza diretta in aziende leader nel settore tecnologico. Questa combinazione di conoscenza accademica e pratica reale consente una trasmissione aggiornata dei contenuti, in linea con le esigenze del mercato. Inoltre, la diversità delle specializzazioni all'interno del team garantisce un approccio multidisciplinare che spazia dall'ingegneria del software all'innovazione dei sistemi distribuiti. Ciò favorisce una comprensione completa dell'ambiente digitale, consentendo l'acquisizione di competenze che rispondono alle sfide attuali e alle esigenze emergenti del settore globale.



66

Accederai ad un piano di studi progettato da un personale docente altamente qualificato in Sviluppo di Software, che ti garantirà un apprendimento efficiente"

Direttore Ospite Internazionale

Considerato un riferimento internazionale nell'applicazione di strategie digitali, Amey Patil è un prestigioso **Ingegnere di Software** specializzato nell'uso di strumenti tecnologici di ultima generazione.

In questo senso, ha svolto il suo lavoro presso istituzioni riconosciute come **Google** della California. In questo modo, è stato incaricato di gestire molteplici infrastrutture digitali tra cui **Ads Creative Studio, Display & Video 360 o Google Analytics**. Sotto la sua guida, questi prodotti hanno generato ricavi economici significativi che hanno alimentato la **crescita strategica** dell'azienda e migliorato l'**esperienza degli utenti**.

In questo modo, la sua filosofia di lavoro si basa sulla progettazione e l'implementazione di soluzioni **online personalizzate** per le istituzioni con l'obiettivo di ottimizzare sia la loro presenza in rete che il loro rendimento pubblicitario. Grazie a questo, ha guidato numerose **campagne di marketing** che hanno massimizzato il **ritorno sull'investimento**; allo stesso tempo hanno anche rafforzato la reputazione delle organizzazioni nella società.

Inoltre, ha promosso diverse iniziative che promuovono l'**innovazione** nel campo tecnologico. Un esempio di ciò è l'analisi di come l'**Intelligenza Artificiale** consente alle aziende di automatizzare attività complesse come l'elaborazione di grandi volumi di informazioni. In questo modo, ha aiutato un'ampia gamma di aziende a migliorare i loro processi operativi e il processo decisionale informato. Inoltre, questo ha permesso alle aziende di identificare tempestivamente le **tendenze** per creare prodotti esclusivi che si sono posizionati sul mercato.

Nel suo impegno per l'eccellenza, Amey Patil partecipa come relatore a **congressi scientifici internazionali**. Così, condivide le sue conoscenze complete su argomenti come l'adozione di **Metodologie Agili**, sistemi per garantire la **Qualità nelle Applicazioni Informatiche** e gli ultimi progressi nella gestione di **Database** in ambienti aziendali.



Dott. Patil, Amey

- Direttore di Ingegneria del Software di Google in California, USA
- Direttore di Ingegneria presso Dell Technologies, California
- Sviluppatore Software presso Mealpit, California
- Ingegnere di Garanzia della Qualità presso Amdocs, Cipro
- Specialista di Google Ads e Google Analytics
- Master in Scienze dell'ingegneria presso l'Università del Maryland
- Laurea in Telecomunicazioni presso Università di Mumbai

“

*Grazie a TECH potrai
apprendere dai migliori
professionisti del mondo”*

09

Titolo

Il Master in Sviluppo di Software garantisce, oltre alla preparazione più rigorosa e aggiornata, il conseguimento della qualifica di Master rilasciato da TECH Global University.



66

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

Questo programma permetterà di ottenere il titolo di studio privato di **Master in Sviluppo di Software** rilasciato da **TECH Global University**, la più grande Università digitale del mondo.

TECH Global University, è un'Università Ufficiale Europea riconosciuta pubblicamente dal Governo di Andorra ([bollettino ufficiale](#)). Andorra fa parte dello Spazio Europeo dell'Istruzione Superiore (EHEA) dal 2003. L'EHEA è un'iniziativa promossa dall'Unione Europea che mira a organizzare il quadro formativo internazionale e ad armonizzare i sistemi di istruzione superiore dei Paesi membri di questo spazio. Il progetto promuove valori comuni, l'implementazione di strumenti congiunti e il rafforzamento dei meccanismi di garanzia della qualità per migliorare la collaborazione e la mobilità tra studenti, ricercatori e accademici.



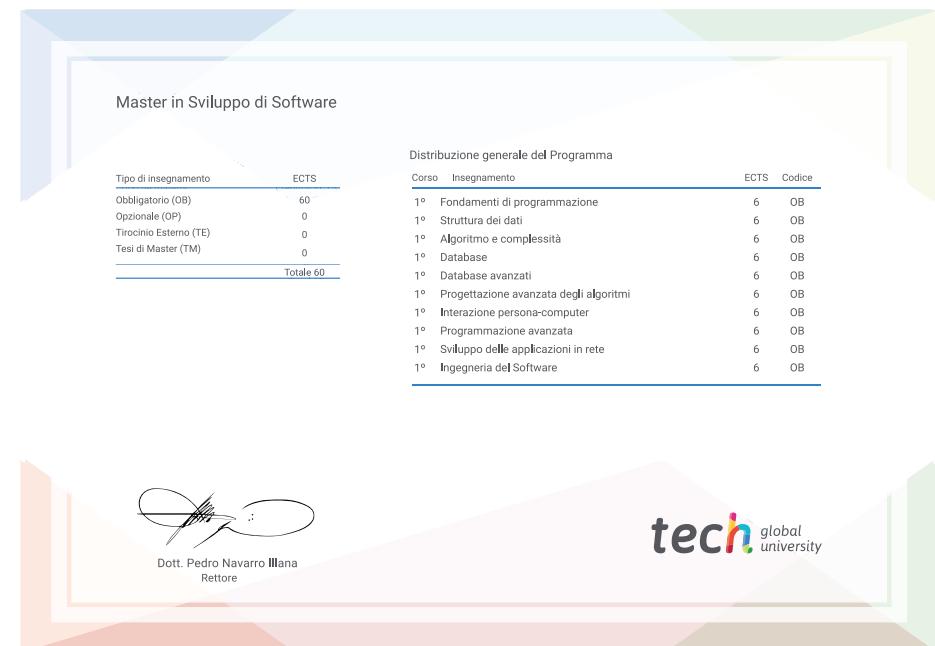
Questo titolo privato di **TECH Global University**, è un programma europeo di formazione continua e aggiornamento professionale che garantisce l'acquisizione di competenze nella propria area di conoscenza, conferendo allo studente che supera il programma un elevato valore curriculare.

Titolo: **Master in Sviluppo di Software**

Modalità: **online**

Durata: **12 mesi**

Accreditamento: **60 ECTS**



*Apostilla dell'Aia. Se lo studente dovesse richiedere che il suo diploma cartaceo sia provvisto di Apostilla dell'Aia, TECH Global University 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



Master
Sviluppo di Software

- » Modalità: online
- » Durata: 12 mesi
- » Titolo: TECH Global University
- » Accreditamento: 60 ECTS
- » Orario: a tua scelta
- » Esami: online

Master

Sviluppo di Software

