

Master

Programmazione da Zero





Master Programmazione da Zero

- » 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-programmazione-zero



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

Metodologia di studio

pag. 34

07

Personale docente

pag. 44

08

Titolo

pag. 48

01

Presentazione del programma

La Programmazione è diventata un'abilità essenziale nel XXI secolo, svolgendo un ruolo chiave nella trasformazione digitale delle società e delle economie. Secondo un rapporto dell'OCSE, oltre il 60% dei lavori attuali richiede competenze digitali avanzate, tra le quali la programmazione è una delle più richieste. Considerando che è un campo essenziale nell'era digitale, TECH ha ideato questo programma che affronterà i fondamenti di tale disciplina e la sua applicazione in linguaggi di uso globale. Attraverso una metodologia 100% online, gli specialisti scopriranno come strutturare algoritmi, gestire dati e sviluppare progetti funzionali, costruendo nel contempo una solida base per avanzare ulteriormente nel mondo tecnologico.



66

La programmazione non è mai stata così accessibile. Grazie al Master di TECH padroneggerai linguaggi come Python e JavaScript da qualsiasi luogo, grazie alla metodologia 100% online. Iscriviti subito!

La Programmazione è al centro della trasformazione digitale che stanno vivendo le società moderne. Dallo sviluppo di applicazioni mobili fino all'automazione dei processi aziendali, saper programmare è un'abilità fondamentale per avanzare in un ambiente di lavoro in continua evoluzione. In questo senso, i professionisti che padroneggiano questo campo non solo troveranno migliori opportunità di crescita, ma potenzieranno anche la capacità di adattarsi e prosperare in un mondo in cui la tecnologia è in continua evoluzione.

Consapevoli di questa necessità, TECH ha progettato il Master in Programmazione per Principianti, una qualifica completa che fornirà le conoscenze più aggiornate relative a questo campo. Il piano di studi, progettato da esperti del settore, comprenderà le basi di algoritmi e strutture dati, fino all'uso pratico di linguaggi come *Python*, *JavaScript* e *HTML*. Comprenderà anche moduli specializzati in web design, sviluppo di applicazioni e gestione di database. In questo modo, gli specialisti acquisiranno conoscenze tecniche e svilupperanno capacità analitiche e creative per risolvere i problemi attraverso la programmazione. Tutto questo, con un approccio pratico che garantirà l'implementazione dei concetti appresi.

Specializzandosi in questo settore, gli studenti non solo troveranno nuove opportunità nel settore tecnologico, ma si distingueranno anche in settori come quello finanziario, sanitario o logistico, dove la Programmazione svolge un ruolo chiave. Inoltre, saranno formati per guidare progetti di sviluppo, automatizzare i processi e ottimizzare le risorse, aumentando il loro profilo lavorativo e facilitando l'accesso a opportunità altamente competitive.

Infine, la modalità 100% online di questo programma offrirà la flessibilità necessaria per conciliare l'apprendimento con altri impegni personali o professionali. TECH metterà a disposizione degli studenti una piattaforma innovativa accessibile 24 ore su 24. Inoltre, la metodologia *Relearning* ottimizzerà la formazione ripetendo concetti chiave in contesti diversi, facilitando un'assimilazione progressiva ed efficiente del contenuto.

Questo **Master in Programmazione da Zero** possiede il programma educativo più completo e aggiornato del mercato. Le caratteristiche principali del corso sono:

- ◆ Sviluppo di casi pratici presentati da esperti in Programmazione
- ◆ Contenuti grafici, schematici ed eminentemente pratici che forniscono informazioni scientifiche e pratiche sulle discipline essenziali per l'esercizio della professione
- ◆ Esercizi pratici con cui è possibile valutare sé stessi per migliorare l'apprendimento
- ◆ Particolare enfasi sulle metodologie innovative
- ◆ Lezioni teoriche, domande all'esperto e/o al tutor, forum di discussione su questioni controverse e compiti di riflessione individuale
- ◆ Disponibilità di accesso ai contenuti da qualsiasi dispositivo fisso o portatile con una connessione internet

“

Dagli algoritmi di base allo sviluppo di applicazioni, questo programma ti porterà da zero ad esperto in pochissimo tempo. Diventerai il professionista di cui l'industria tecnologica ha bisogno!"

“

Immergiti nel mondo della programmazione con un piano di studi unico. Padroneggerai Python, JavaScript e lo sviluppo web da zero con il Master più completo del mercato”

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

Con il supporto di docenti esperti in tecnologia, questa qualifica ti guiderà passo dopo passo per padroneggiare i linguaggi più richiesti del mercato. Inizia subito e costruisci un futuro migliore!

Scoprirai un approccio innovativo che combina teoria e progetti reali in questo Master. Iscriviti subito e dai vita alle tue idee con codice!



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".



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

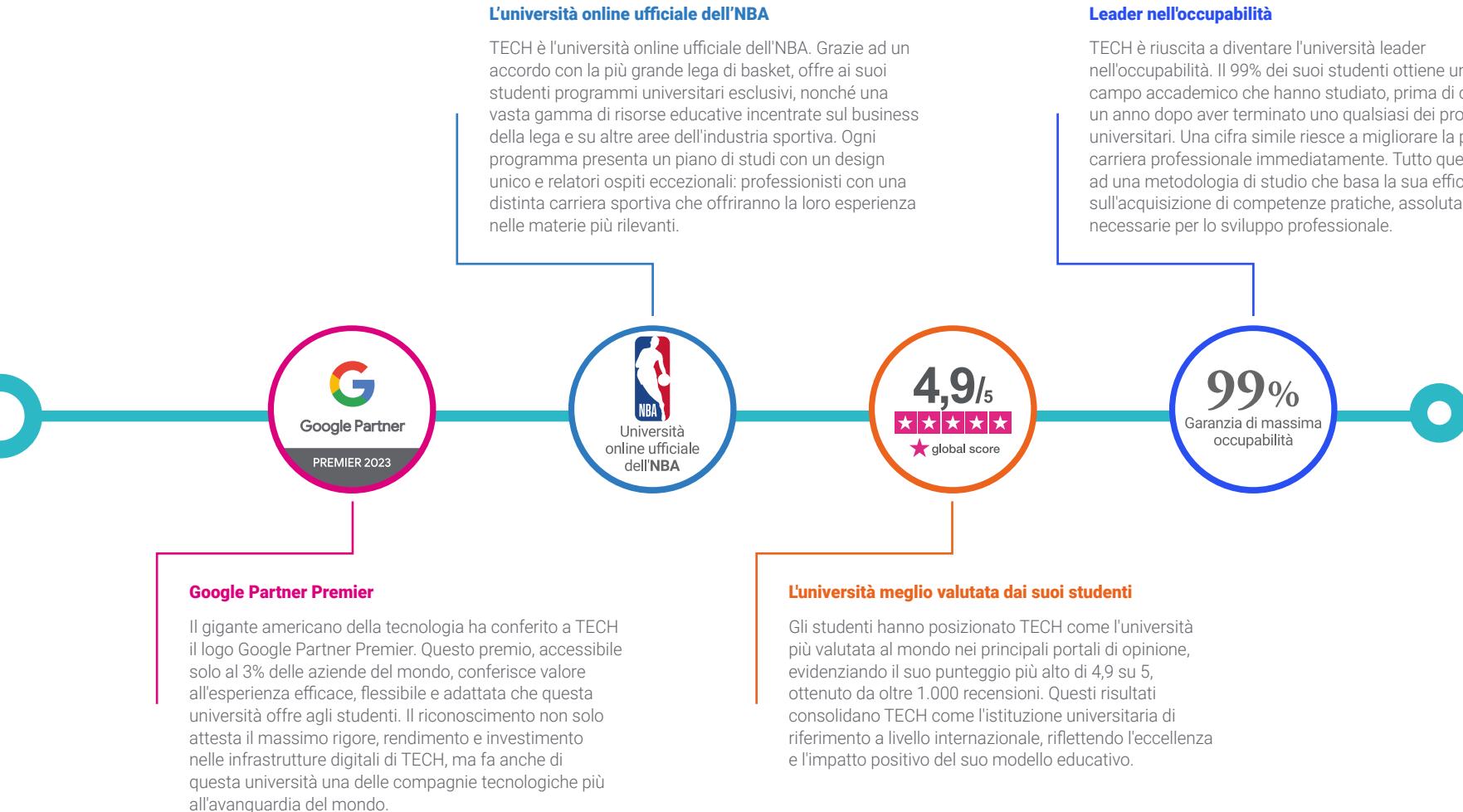
Il personale docente di TECH è composto da oltre 6.000 docenti di massimo prestigio internazionale. Professori, ricercatori e dirigenti di multinazionali, tra cui Isaiah Covington, allenatore dei Boston Celtics; Magda Romanska, ricercatrice principale presso MetaLAB ad Harvard; Ignacio Wistumbá, presidente 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

Nel corso del programma, i professionisti affronteranno le basi di algoritmi, strutture dati e logica di programmazione, per poi approfondire linguaggi come *Python*, *JavaScript* e *HTML*. Inoltre, approfondiranno aree chiave come il web design, lo sviluppo di applicazioni e la gestione dei database, offrendo una visione completa del mondo della programmazione. Grazie ad un approccio graduale e strutturato, questo piano di studi garantirà non solo l'acquisizione di conoscenze tecniche, ma anche lo sviluppo delle competenze analitiche e creative indispensabili in un ambiente di lavoro competitivo.

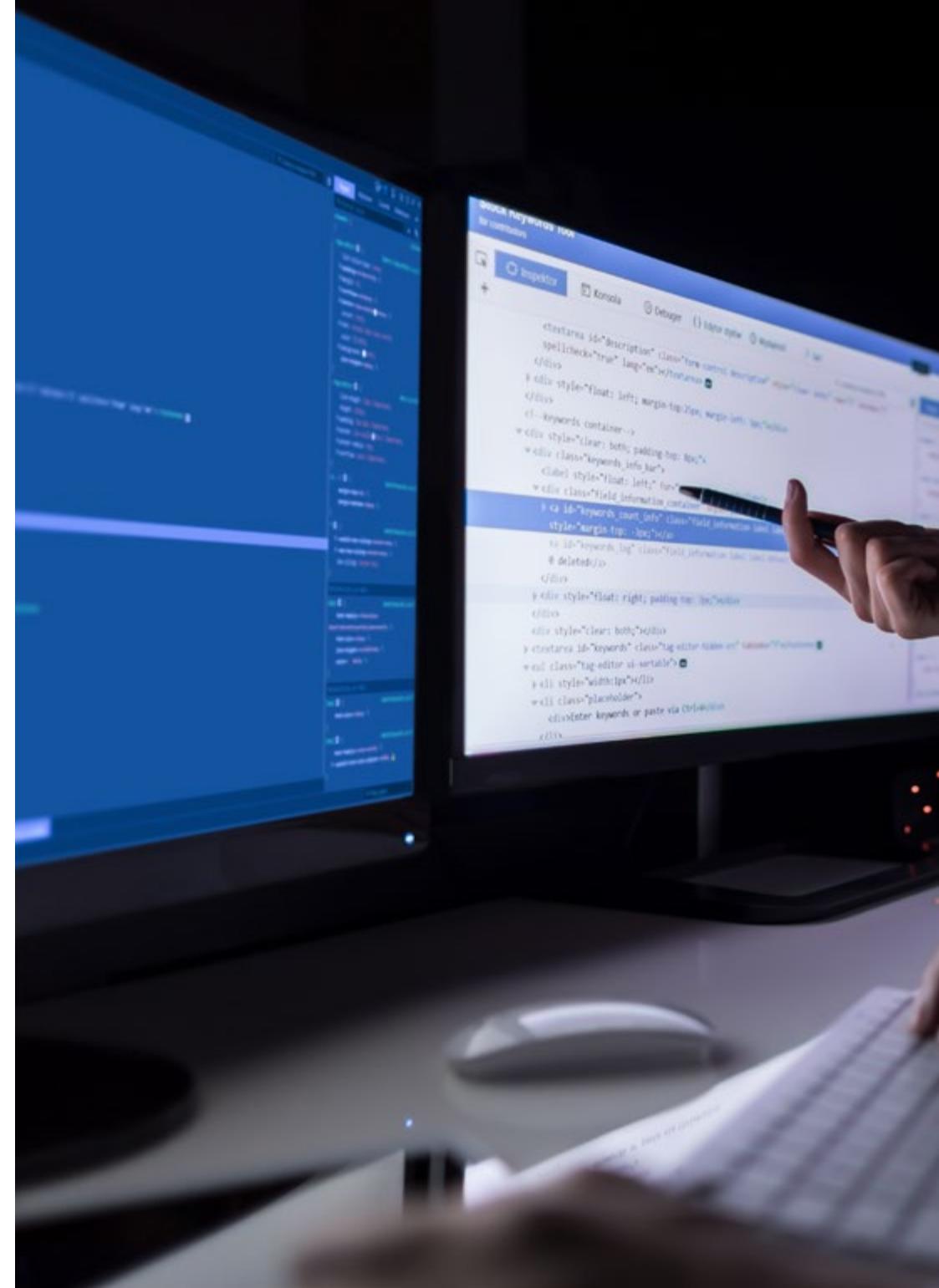


66

Questo corso post-laurea si presenta come l'opzione ideale per iniziare la tua carriera nel settore tecnologico e acquisire gli strumenti necessari per eccellere in qualsiasi ambito professionale relativo alla Programmazione"

Modulo 1. Programmazione e Sviluppo di Software da Zero

- 1.1. Software e hardware: Relazione e differenze
 - 1.1.1. Software
 - 1.1.2. Differenze tra software e hardware
 - 1.1.3. Relazione tra software, hardware e programmazione
- 1.2. Programmazione: Aspetti fondamentali
 - 1.2.1. Programmazione
 - 1.2.2. Obiettivi e applicazioni
 - 1.2.3. Programmi, codice sorgente, compilazione ed esecuzione
 - 1.2.4. Errori: Errori di sintassi, di esecuzione, logici
- 1.3. Programmazione da zero
 - 1.3.1. Struttura di un programma
 - 1.3.2. Algoritmi: Struttura ed esempi
 - 1.3.3. Relazione tra algoritmi e programmi
 - 1.3.4. Risoluzione dei problemi tramite algoritmi
- 1.4. Paradigmi e tipi di linguaggi di programmazione
 - 1.4.1. Paradigmi programmazione
 - 1.4.1.1. Paradigmi imperativi
 - 1.4.1.2. Paradigma orientato agli oggetti
 - 1.4.1.3. Paradigma funzionale
 - 1.4.1.4. Paradigma dichiarativo
 - 1.4.2. Linguaggi di basso livello vs alto livello
 - 1.4.3. Linguaggi compilati vs interpretati
- 1.5. Traduzione di linguaggi di programmazione
 - 1.5.1. Il compilatore: Processo di compilazione
 - 1.5.2. L'Interprete: Processo di interpretazione
 - 1.5.3. Differenze tra compilazione e interpretazione
- 1.6. Bit, operazioni binarie e porte logiche
 - 1.6.1. Bit: Rappresentazione binaria
 - 1.6.2. Operazioni di base con i bit: AND, OR, XOR, NOT
 - 1.6.3. Conversione tra binario e decimale
 - 1.6.4. Porte logiche: OR, AND, XOR, NOT, NOR e NAND





- 1.7. Progettazione di algoritmi da zero
 - 1.7.1. Progettazione di un algoritmo da zero
 - 1.7.2. Algoritmi sequenziali, condizionali e ripetitivi
 - 1.7.3. Algoritmi ricorsivi e il loro confronto con algoritmi iterativi
- 1.8. Elementi del programma: Componenti e struttura
 - 1.8.1. Input e output di dati
 - 1.8.2. Variabili e costanti: Uso
 - 1.8.3. Elaborazione e trattamento dei dati
 - 1.8.4. Funzioni e procedure di base con diagrammi di flusso
- 1.9. Strutture di controllo con diagrammi di flusso
 - 1.9.1. Strutture di controllo: Ruolo nella programmazione
 - 1.9.2. Strutture condizionali: Esempi con diagrammi di flusso
 - 1.9.3. Strutture ripetitive: Esempi con diagrammi di flusso
- 1.10. Ciclo di vita e modelli di sviluppo software
 - 1.10.1. Ciclo di vita del software: Fasi
 - 1.10.2. Modelli di sviluppo: Cascata, iterativo e agile
 - 1.10.3. Testing e manutenzione nello sviluppo di software

Modulo 2. Sviluppo Backend I: Python da Zero

- 2.1. Python da zero: Installazione
 - 2.1.1. Linguaggio Python: Caratteristiche
 - 2.1.2. Installazione di Python su Windows, macOS e Linux
 - 2.1.3. Configurazione dell'ambiente di sviluppo: IDE ed editori di codice
 - 2.1.4. Primo programma in Python: "Hello World"
- 2.2. Sintassi e variabili in Python
 - 2.2.1. Strutture del codice in Python: Indentazione
 - 2.2.2. Commenti in Python
 - 2.2.3. Variabili e tipi di dati in Python
 - 2.2.4. Operazioni aritmetiche e logiche in Python
- 2.3. Controllo di flusso: Condizionali
 - 2.3.1. Strutture di controllo
 - 2.3.2. Condizionali: if, elif, else
 - 2.3.3. Condizionale ternario

- 2.4. Loop in Python
 - 2.4.1. Uso di loop nella programmazione
 - 2.4.2. Loop "for" e "while"
 - 2.4.3. Controllo del flusso in loop: break e continue
 - 2.4.4. Loop nidificati
- 2.5. Funzioni in Python
 - 2.5.1. Funzioni in Python: Utilità
 - 2.5.2. Parametri e argomenti delle funzioni
 - 2.5.3. Valori di ritorno
 - 2.5.4. Funzioni predefinite vs. funzioni create dall'utente
- 2.6. Liste e tuple in Python
 - 2.6.1. Creazione e utilizzo di liste in Python
 - 2.6.2. Operazioni comuni con liste: Aggiungere, eliminare, modificare
 - 2.6.3. Tuple: Differenze con le liste
 - 2.6.4. Iterazione su liste e tuple
- 2.7. Dizionari e set in Python
 - 2.7.1. Dizionari: Chiave-Valore
 - 2.7.2. Metodi per manipolare i dizionari
 - 2.7.3. Set: Uso
 - 2.7.4. Confronto di dizionari e set
- 2.8. Gestione dei file in Python da zero
 - 2.8.1. Apertura e chiusura di file
 - 2.8.2. Modalità di apertura: Lettura, scrittura e aggiunta
 - 2.8.3. Lettura e scrittura di file di testo
- 2.9. Gestione di errori ed eccezioni
 - 2.9.1. Tipi di eccezioni
 - 2.9.2. Utilizzo di Try, Except per la gestione degli errori
 - 2.9.3. Creazione di eccezioni personalizzate
- 2.10. Buone pratiche e debug in Python
 - 2.10.1. Debugging: Propositi
 - 2.10.2. Tecniche di debug: Utilizzo di print e breakpoint
 - 2.10.3. Buone pratiche nella scrittura del codice

Modulo 3. Sviluppo Backend II - Algoritmi e Strutture di Dati con Python da Zero

- 3.1. Algoritmi di ricerca in strutture di dati
 - 3.1.1. Scopo degli algoritmi di ricerca nelle strutture di dati
 - 3.1.2. Ricerca lineare: Implementazione e casi d'uso
 - 3.1.3. Ricerca binaria: Esempi
 - 3.1.4. Confronto di efficienza: Ricerca lineare vs binaria
- 3.2. Algoritmi di ordinamento in strutture di dati (I). Tecniche di base per l'ordinamento Bubble Sort e Insertion Sort
 - 3.2.1. Ordinamento a bolla (Bubble Sort): Implementazione e analisi
 - 3.2.2. Ordinamento per inserimento (Insertion Sort): Implementazione e casi d'uso
 - 3.2.3. Confronto tra Bubble Sort e Insertion Sort
- 3.3. Algoritmi di ordinamento in strutture di dati (II). Tecniche avanzate di ordinamento Selection Sort, Merge Sort e Quick Sort
 - 3.3.1. Selection Sort: Implementazione e analisi
 - 3.3.2. Merge Sort: Implementazione
 - 3.3.3. Quick Sort: Implementazione
 - 3.3.4. Confronto di efficienza tra algoritmi di ordinamento
- 3.4. Algoritmi di ricerca ricorsiva in strutture di dati
 - 3.4.1. Ricorsione: Uso
 - 3.4.2. Ricorsione diretta e indiretta
 - 3.4.3. Algoritmi ricorsivi: Factorial e Fibonacci. Esempi
- 3.5. Complessità degli algoritmi di ricerca nelle strutture dati
 - 3.5.1. Complessità algoritmica: Misurazione dell'efficienza
 - 3.5.2. Notazione Big-O
 - 3.5.3. Analisi della complessità negli algoritmi di ricerca e ordinamento
- 3.6. Strutture di dati avanzate
 - 3.6.1. Alberi: Terminologia
 - 3.6.2. Alberi binari: Operazioni
 - 3.6.3. Alberi di ricerca binari (BST): Ricerca, inserimento ed eliminazione
- 3.7. Algoritmi di grafi
 - 3.7.1. Grafi: Rappresentazione
 - 3.7.2. Algoritmi di grafi: DFS e BFS
 - 3.7.3. Confronto tra DFS e BFS

- 3.8. Programmazione dinamica
 - 3.8.1. Programmazione dinamica: Applicazioni
 - 3.8.2. Differenze tra programmazione dinamica e ricorsione
 - 3.8.3. Ottimizzazione tramite programmazione dinamica
- 3.9. Tecniche di ottimizzazione degli algoritmi di ricerca dei dati
 - 3.9.1. Importanza dell'ottimizzazione algoritmica della ricerca di dati
 - 3.9.2. Tecniche di ottimizzazione: Memorizzazione (Memoization)
 - 3.9.3. *Divide and Conquer*: Dividi e vincerali
- 3.10. Altri algoritmi in Python
 - 3.10.1. Algoritmi di permutazione e combinazione
 - 3.10.2. Algoritmi di hashing di base
 - 3.10.3. Algoritmi di conteggio e generazione di sottoinsiemi

Modulo 4. Sviluppo Backend III - Flask, Creazione di API e Architettura di Base da Zero

- 4.1. Flask come Framework Backend
 - 4.1.1. Framework Backend: Scopo
 - 4.1.2. Flask: Caratteristiche
 - 4.1.3. Preparazione dell'ambiente di sviluppo e installazione di Flask
 - 4.1.4. Primo progetto con Flask: "Hello World"
- 4.2. Percorsi e richieste HTTP in Flask
 - 4.2.1. Percorso: Funzionamento in un'applicazione web
 - 4.2.2. Metodi HTTP in Flask: GET, POST, PUT e DELETE
 - 4.2.3. Percorsi in Flask con parametri e dati
 - 4.2.4. Organizzazione dei percorsi in un progetto
- 4.3. Controller e gestione delle risposte in Flask
 - 4.3.1. Controller: Ruoli e responsabilità
 - 4.3.2. Tipi di risposte in Flask: Testo, JSON e HTML
 - 4.3.3. Creazione di driver per API in Flask
 - 4.3.4. Operazioni CRUD su controller
- 4.4. API RESTful
 - 4.4.1. API RESTful: Principi
 - 4.4.2. Metodi HTTP e convenzioni nelle API RESTful
 - 4.4.3. Creazione di API RESTful in Flask
 - 4.4.4. Progettazione di API delle attività con operazioni CRUD
- 4.5. Database e Flask con SQLite
 - 4.5.1. Database nelle applicazioni web
 - 4.5.2. Connessione a SQLite nei progetti Flask
 - 4.5.3. Creazione di tabelle e modelli utilizzando SQLAlchemy
 - 4.5.4. Query CRUD in SQLite per la gestione dei dati
- 4.6. Autenticazione e sicurezza di base nelle API
 - 4.6.1. Autenticazione e autorizzazione nelle API
 - 4.6.2. Creazione di un sistema di autenticazione dell'utente
 - 4.6.3. Utilizzo di token per l'autenticazione in Flask
 - 4.6.4. Protezione dei percorsi e dei dati utente nelle API
- 4.7. Validazione dei dati e gestione degli errori
 - 4.7.1. Gestione di errori ed eccezioni in Flask
 - 4.7.2. Convalida dei dati nelle richieste API
 - 4.7.3. Creazione di messaggi di errore personalizzati
 - 4.7.4. Strategie di convalida e gestione degli errori in CRUD
- 4.8. Strutturazione di API scalabili
 - 4.8.1. Organizzazione e struttura di un progetto Flask scalabile
 - 4.8.2. Modularizzazione e separazione delle responsabilità nelle API
 - 4.8.3. Ottimizzazione di base delle API per prestazioni e scalabilità
 - 4.8.4. Strategia organizzativa per i grandi progetti
- 4.9. Comunicazione in tempo reale con WebSockets
 - 4.9.1. WebSockets: Applicazioni
 - 4.9.2. Implementazione di WebSockets in Flask con Flask-SocketIO
 - 4.9.3. Comunicazione in tempo reale nelle applicazioni Flask
- 4.10. Deploy e manutenzione delle applicazioni
 - 4.10.1. Preparazione delle applicazioni Flask per la produzione
 - 4.10.2. Deploy su piattaforme popolari come Heroku e Render
 - 4.10.3. Utilizzo di Docker per la distribuzione su container
 - 4.10.4. Monitoraggio e manutenzione delle applicazioni backend

Modulo 5. Programmazione Orientata a Oggetti e Modelli di Progettazione da Zero

- 5.1. Programmazione Orientata agli Oggetti (OOP) da Zero
 - 5.1.1. Programmazione Orientata agli Oggetti
 - 5.1.2. Differenze tra OOP e programmazione strutturata
 - 5.1.3. Elementi OOP: Classi, oggetti, metodi e attributi
- 5.2. Classi e oggetti in Python
 - 5.2.1. Creazione di classi e oggetti in Python
 - 5.2.2. Attributi di istanza e classe
 - 5.2.3. Metodi speciali (init, str, repr, ecc.)
 - 5.2.4. Metodi statici e di classe: Usi
- 5.3. Incapsulamento e astrazione nelle classi
 - 5.3.1. Incapsulamento: Utilità
 - 5.3.2. Modificatori di accesso in Python
 - 5.3.2.1. Pubblico, protetto e privato
 - 5.3.3. Astrazione: Nascondere i dettagli e migliorare la semplicità
 - 5.3.4. Uso di proprietà (@property) per il controllo degli accessi
- 5.4. Eredità in Python: Utilità in OOP
 - 5.4.1. Eredità: Utilità in OOP
 - 5.4.2. Creazione di classi derivate ed ereditarietà multipla in Python
 - 5.4.3. Metodi e attributi ereditati e sovraccarico sull'eredità
 - 5.4.4. Gerarchie di classe e gestione delle classi base
- 5.5. Polimorfismo e sovraccarico in Python
 - 5.5.1. Polimorfismo: Duck Typing
 - 5.5.2. Polimorfismo con classi e metodi in Python
 - 5.5.3. Sovraccarico e sovrascrittura dei metodi in Python
 - 5.5.4. Il polimorfismo nella progettazione del software: Applicazioni e vantaggi
- 5.6. Relazioni tra classi e progettazione di strutture complesse
 - 5.6.1. Tipi di relazioni: Associazione, aggregazione e composizione
 - 5.6.2. Differenze tra aggregazione e composizione: Esempi
 - 5.6.3. Progettazione di strutture complesse utilizzando relazioni tra classi

- 5.7. Modelli di progettazione e principi SOLID
 - 5.7.1. Rilevanza dei modelli di progettazione
 - 5.7.2. Applicazione di modelli di progettazione in progetti OOP: Vantaggi
 - 5.7.3. Classificazione dei modelli di progettazione
 - 5.7.4. Principi SOLID e la loro importanza nel design orientato agli oggetti
- 5.8. Modelli di design creativi
 - 5.8.1. Scopo dei modelli di design creativi
 - 5.8.2. Pattern Singleton
 - 5.8.3. Modello Factory/Factory Method
 - 5.8.4. Modello Builder
- 5.9. Modelli di design strutturali
 - 5.9.1. Scopo dei modelli di progettazione strutturale
 - 5.9.2. Modello Adapter
 - 5.9.3. Modello Decorador
 - 5.9.4. Modello Facade
- 5.10. Modelli di progettazione comportamentale
 - 5.10.1. Modelli di comportamento: applicazioni
 - 5.10.2. Modello Observer
 - 5.10.3. Modello Strategy

Modulo 6. Frontend I - HTML e CSS da Zero

- 6.1. HTML da zero
 - 6.1.1. HTML: Scopo nello sviluppo web
 - 6.1.2. Struttura di un documento HTML: DOCTYPE, <html>, <head>, <body>
 - 6.1.3. Etichette semantiche e di contenuto: <header>, <nav>, <section>, <footer>
 - 6.1.4. Elementi essenziali: Paragrafi (<p>), liste (,), link (<a>), immagini ()
 - 6.1.5. Buone pratiche di HTML
- 6.2. Elementi di testo e multimediali in HTML
 - 6.2.1. Etichette di testo essenziali: <p>, intestazioni, elenchi, grassetto e corsivo
 - 6.2.2. Inserimento di multimedia: Attributi di , <audio>, <video>
 - 6.2.3. Attributi essenziali per accessibilità (alt, aria-label)
- 6.3. Moduli in HTML
 - 6.3.1. Struttura e componenti dei moduli: <form>, <input>, <label>, <button>
 - 6.3.2. Tipi di biglietti: Testo, email, password, pulsanti di invio
 - 6.3.3. Validazione HTML5: Convalida dei campi sul lato client
 - 6.3.4. Moduli con convalida di base: Esempi

- 6.4. CSS da zero
 - 6.4.1. Linguaggio CSS da zero: Uso e relazione con HTML
 - 6.4.2. Sintassi di CSS: Selettori, proprietà e valori
 - 6.4.3. Applicazione di stili online, interni ed esterni
 - 6.4.4. Selettori avanzati: Tipo, classe ID, pseudo-classi (:hover, :focus)
- 6.5. modello a scatola (Box Model) in CSS
 - 6.5.1. Box Model: Importanza in CSS
 - 6.5.2. Proprietà chiave: margin, padding, border, width, height
 - 6.5.3. Utilizzo di box-sizing per un controllo preciso del box model
 - 6.5.4. La progettazione applicata al box model: Esempi
- 6.6. Tipografia e stili di testo in CSS
 - 6.6.1. Proprietà di colori e font: color, font-family, font-size
 - 6.6.2. Stili di testo avanzati: Grassetto, corsivo, allineamento (text-align)
 - 6.6.3. Spaziatura e separazione del testo: line-height, letter-spacing
 - 6.6.4. Unità di misura in CSS (px, em, rem) e il loro uso nella tipografia
- 6.7. Progettazione con CSS - Flexbox
 - 6.7.1. Flexbox: Propositi
 - 6.7.2. Proprietà di Flexbox: justify-content, align-items, flex-direction
 - 6.7.3. Distribuzione degli elementi e allineamento in Flexbox
 - 6.7.4. Esempi di layout con Flexbox
- 6.8. CSS Grid e Responsive Design con CSS
 - 6.8.1. CSS Grid: Righe, colonne e aree
 - 6.8.2. Media Queries: Struttura e applicazione su diversi dispositivi
 - 6.8.3. Responsive design per dispositivi mobili, tablet e desktop
 - 6.8.4. Regolazione della tipografia e delle unità fluide in responsive design
- 6.9. Animazioni e transizioni in CSS
 - 6.9.1. Transizioni: Proprietà transition, effetti su :hover
 - 6.9.2. Animazioni con CSS: Uso di @keyframes, animazioni di base
 - 6.9.3. Tecniche per lasciare transizioni e animazioni sul web
- 6.10. Accessibilità Web in Design
 - 6.10.1. L'accessibilità del web: Importanza
 - 6.10.2. Progettazione di siti accessibili. Best practice
 - 6.10.3. Etichette aria e strumenti di convalida dell'accessibilità

Modulo 7. Frontend II – JavaScript da Zero

- 7.1. JavaScript da zero
 - 7.1.1. Linguaggio JavaScript
 - 7.1.2. Integrazione di JavaScript in HTML
 - 7.1.3. Primo programma in JavaScript: "Hello World"
- 7.2. Variabili e tipi di dati in JavaScript
 - 7.2.1. Dichiarazione di variabili con var, let e const
 - 7.2.2. Tipi di dati: Numeri, array, booleani
 - 7.2.3. Conversione tra tipi di dati
- 7.3. Strutture di controllo in JavaScript
 - 7.3.1. Condizionali: if, else, else if
 - 7.3.2. Loop: for, while, do...while
 - 7.3.3. Switch-case: Alternativa a più condizionali
 - 7.3.4. Break e continue in loop
- 7.4. Funzioni in JavaScript
 - 7.4.1. Dichiarazioni di funzioni
 - 7.4.2. Parametri, valori di ritorno e scope
 - 7.4.3. Funzioni freccia (=>) e funzioni anonime
 - 7.4.4. Callback e ricorsività nelle funzioni
- 7.5. Manipolazione del DOM (Document Object Model) con JavaScript
 - 7.5.1. DOM: Struttura di un documento HTML
 - 7.5.2. Selezione degli elementi del DOM (getElementById, querySelector)
 - 7.5.3. Gestione degli elementi: Modifica testo, stili e attributi
 - 7.5.4. Eventi: click, input, submit, ecc.
- 7.6. Correzioni e oggetti in JavaScript
 - 7.6.1. Dichiarazione e utilizzo di accordi
 - 7.6.2. Metodi di arrangiamento comuni: push, pop, map, filter
 - 7.6.3. Creazione e utilizzo di oggetti
 - 7.6.4. Iterazione su array e oggetti
- 7.7. Promesse e asincronia in JavaScript
 - 7.7.1. Asincronia e utilizzo di callback in JavaScript
 - 7.7.2. Promise in JavaScript: Creazione e gestione
 - 7.7.3. Utilizzo di async e await in JavaScript

- 7.8. API e Fetch in JavaScript
 - 7.8.1. API in JavaScript: Propositi
 - 7.8.2. Consumo di API REST con fetch
 - 7.8.3. Gestione degli errori e degli stati delle richieste
- 7.9. Archiviazione locale nei browser web
 - 7.9.1. Local Storage e Session Storage nei servizi e nelle applicazioni web
 - 7.9.2. Archiviazione e recupero dei dati in Local Storage
 - 7.9.2. IndexedDB come database per il browser
 - 7.9.3. Gestione dei cookie in JavaScript
 - 7.9.4. Archiviazione nel browser: Esempi
- 7.10. Buone pratiche in JavaScript e strumenti per gli sviluppatori
 - 7.10.1. Codice in JavaScript: Best practice
 - 7.10.2. Utilizzo di strumenti di sviluppo del browser in JavaScript
 - 7.10.3. Debug e gestione degli errori in JavaScript

Modulo 8. Frontend III - React.js da Zero

- 8.1. React.js da zero
 - 8.1.1. React.js come libreria per lo sviluppo di applicazioni web
 - 8.1.2. Componenti e Virtual DOM in React.js: Architettura e funzionamento
 - 8.1.3. Installazione e configurazione con i framework NextJS
 - 8.1.4. Primo componente in React: "Hello World"
- 8.2. JavaScript XML o JSX, e componenti in React
 - 8.2.1. JSX: Sintassi e caratteristiche
 - 8.2.2. Creazione di componenti funzionali in React.js
 - 8.2.3. Utilizzo di props per il passaggio dei dati tra i componenti
 - 8.2.4. Componenti funzionali rispetto ai componenti di classe per lo sviluppo in React.js
- 8.3. Stato ed eventi in React.js
 - 8.3.1. Stato dei componenti in React
 - 8.3.2. Uso di useState per la gestione dello stato
 - 8.3.3. Gestione degli eventi in React.js: onClick, onChange, ecc.
 - 8.3.4. Esempi di gestione dello stato e degli eventi in React.js
- 8.4. Ciclo di vita dei componenti ed effetti in React
 - 8.4.1. Ciclo di vita dei componenti in React
 - 8.4.2. Uso di useEffect per gestire gli effetti in React
 - 8.4.3. Componenti con montaggio, aggiornamento e smontaggio in React
- 8.5. Routing con React Router
 - 8.5.1. SPA (Single Page Applications) e routing nelle applicazioni web
 - 8.5.2. Installazione e configurazione React Router
 - 8.5.3. Creazione di percorsi e navigazione tra le pagine con React Router
- 8.6. Moduli e convalida in React
 - 8.6.1. Creazione di moduli interattivi in React
 - 8.6.2. Gestione degli input dell'utente e invio dei dati in React
 - 8.6.3. Validazione di moduli in tempo reale su React
- 8.7. Consumo di API in React
 - 8.7.1. Consumo di API con fetch e axios in React
 - 8.7.2. Gestione degli stati di carico, successo ed errore in React
 - 8.7.3. Aggiornamento dei componenti in base ai dati API in React
- 8.8. Componenti riutilizzabili e librerie esterne in React
 - 8.8.1. Componenti riutilizzabili in React
 - 8.8.2. Creazione di componenti riutilizzabili in React
 - 8.8.3. Utilizzo di librerie esterne come Material UI e Bootstrap in React
- 8.9. Gestione globale dello stato in React
 - 8.9.1. Gestione globale dello stato con opzioni native: Context API e Hooks personalizzati
 - 8.9.2. Librerie esterne per la gestione dei dati
 - 8.9.3. Confronto di approcci per la gestione globale dello stato: Esempi
- 8.10. Deploy e ottimizzazione delle applicazioni React
 - 8.10.1. Preparazione delle applicazioni React per la produzione
 - 8.10.2. Deploy su piattaforme come Netlify e Vercel
 - 8.10.3. Ottimizzazione delle prestazioni: Lazy Loading, memoizzazione, componenti del server e code splitting
 - 8.10.4. Monitoraggio e manutenzione delle applicazioni React in produzione: Strumenti e analisi delle prestazioni

Modulo 9. Gestione e Ottimizzazione dei Database da Zero

- 9.1. Database da zero
 - 9.1.1. Database: Tipologie
 - 9.1.2. Database relazionali vs non relazionali
 - 9.1.3. Linguaggi di programmazione SQL e NoSQL

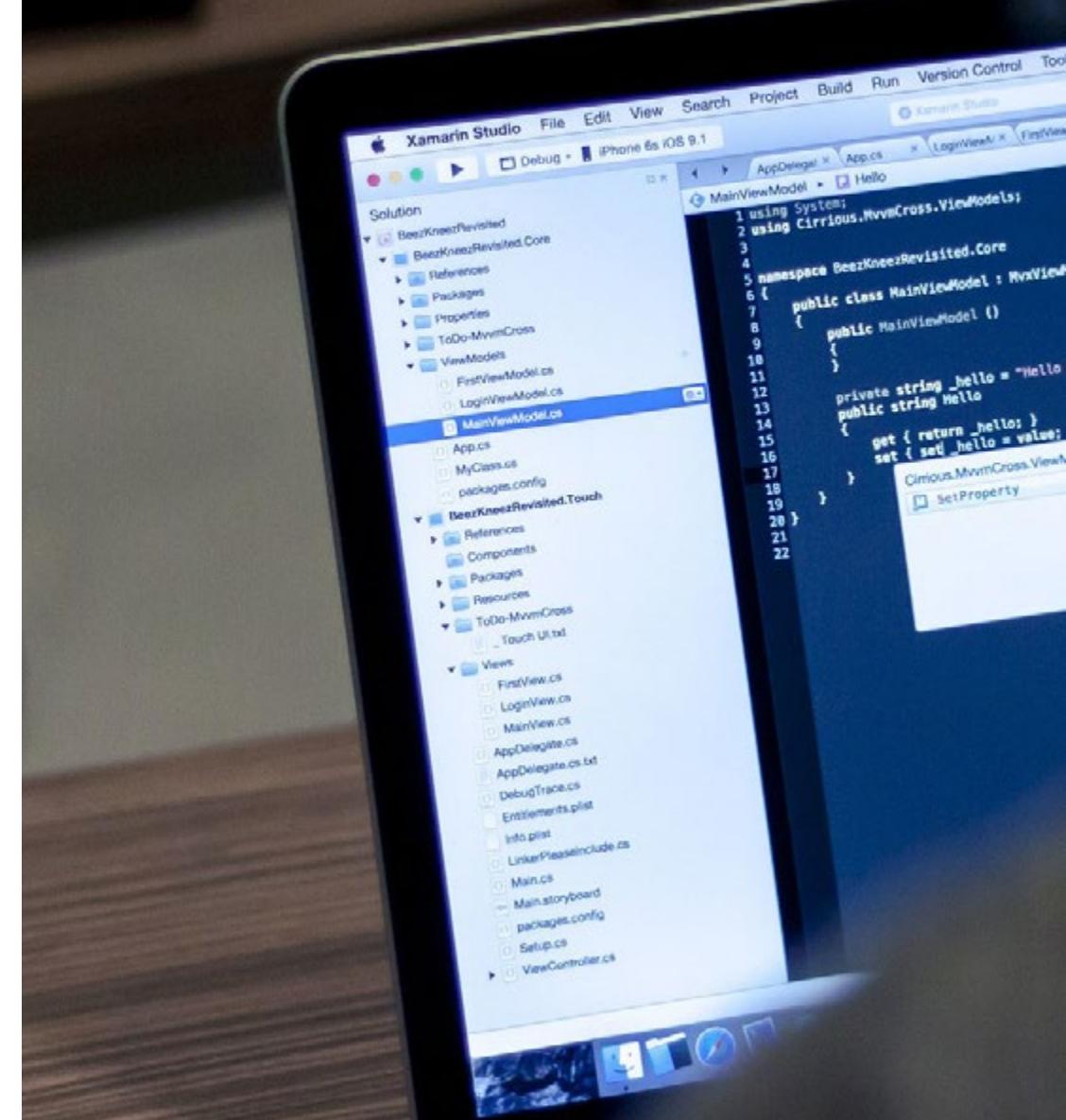
- 9.2. Modellazione di dati relazionali
 - 9.2.1. Modello relazionale del database
 - 9.2.2. Tabelle, righe e colonne in un database relazionale
 - 9.2.3. Chiavi primarie ed esterne: Relazioni tra tabelle
 - 9.2.4. Normalizzazione: 1NF, 2NF, 3NF
- 9.3. Linguaggio SQL: DML e DDL
 - 9.3.1. SQL: Linguaggio query strutturato
 - 9.3.2. Query di creazione e cancellazione: CREATE, DROP
 - 9.3.3. Query SELECT, INSERT, UPDATE e DELETE
 - 9.3.4. Filtraggio e ordinamento dei dati con SQL
- 9.4. Query avanzate in SQL
 - 9.4.1. Joins: INNER JOIN e OUTER JOIN
 - 9.4.2. Sottoquery e query nidificate
 - 9.4.3. Funzioni aggiunte in SQL: SUM, AVG, COUNT
- 9.5. Database NoSQL e MongoDB
 - 9.5.1. Database NoSQL
 - 9.5.2. Confronto tra SQL e NoSQL
 - 9.5.3. MongoDB: Database di documenti
 - 9.5.4. Schemi flessibili in NoSQL
- 9.6. Ottimizzazione dei database
 - 9.6.1. Importanza dell'ottimizzazione delle query
 - 9.6.2. Uso di gli indici nei database relazionali
 - 9.6.3. Ottimizzazione dei database NoSQL
- 9.7. Sicurezza nei database
 - 9.7.1. Sicurezza nei database
 - 9.7.2. Crittografia dei dati sensibili
 - 9.7.3. Gestione degli utenti e dei permessi nei database
 - 9.7.4. Strategie di protezione del database contro gli attacchi
- 9.8. Scalabilità dei database
 - 9.8.1. Scalabilità nei database
 - 9.8.2. Partizione orizzontale e verticale
 - 9.8.3. Replica e clustering nei database

- 9.9. Backup e ripristino dei dati
 - 9.9.1. Importanza del backup nei database
 - 9.9.2. Tecniche di backup automatiche e manuali
 - 9.9.3. Recupero dati in database relazionali e NoSQL
- 9.10. Implementazione di database in progetti
 - 9.10.1. Progettazione di database per un progetto reale
 - 9.10.2. Integrazione di database con applicazioni backend

Modulo 10. Strumenti di Sviluppo da Zero: Linux, Controllo di Versione, CI/CD, Docker e Metodologie Agili

- 10.1. Linux da zero
 - 10.1.1. Linux
 - 10.1.2. Differenze tra Linux e altri sistemi operativi
 - 10.1.3. Distribuzioni popolari di Linux per gli sviluppatori
 - 10.1.4. Configurazione e personalizzazione dell'ambiente di sviluppo
 - 10.1.5. Editor di testo in Linux
- 10.2. Utilizzo del terminale su Linux da zero
 - 10.2.1. Il terminale: Usi e funzionalità
 - 10.2.2. Comandi di navigazione e gestione dei file nel terminale
 - 10.2.3. Autorizzazioni di file e directory su Linux
 - 10.2.4. Reindirizzamento dei comandi e utilizzo di tubi (pipes) per ottimizzare le attività
- 10.3. Controllo delle versioni con Git da zero
 - 10.3.1. Git: Fornitori CLOUD
 - 10.3.2. Creazione e gestione dei repository
 - 10.3.3. Workflow: git init, git add, git commit e git status
 - 10.3.4. Lavoro con i rami: Creazione, fusione e risoluzione dei conflitti
- 10.4. Collaborazione in team con GitHub da zero
 - 10.4.1. GitHub: Repository remoto
 - 10.4.2. Collegamento di un repository locale a GitHub: git remote. Configurazione iniziale
 - 10.4.3. Sincronizzazione con repository remoti
 - 10.4.4. Pull requests e revisione collaborativa del codice

- 10.5. CI/CD (I) - Integrazione continua (CI) con GitHub Actions da zero
 - 10.5.1. Integrazione continua (CI)
 - 10.5.2. Configurazione dei flussi di lavoro in GitHub Actions
 - 10.5.3. Test e implementazione automatizzati
- 10.6. Docker da zero
 - 10.6.1. Docker e container
 - 10.6.2. Installazione e configurazione di Docker
 - 10.6.3. Creazione e gestione di container Docker
 - 10.6.4. Dockerfiles: Creazione di immagini personalizzate
- 10.7. CI/CD (I) - Distribuzione continua (CD) con Docker e GitHub Actions da zero
 - 10.7.1. Distribuzione continua
 - 10.7.2. Configurazione di pipeline CD con Docker e GitHub Actions
 - 10.7.3. Distribuzione automatizzata con Docker Compose
- 10.8. Metodologie agili da zero (I). Principi e valori
 - 10.8.1. Metodologie agili: Principi
 - 10.8.2. Manifesto agile: Valori e principi chiave
 - 10.8.3. Confronto con le metodologie tradizionali: Cascata vs. Agile
- 10.9. Metodologie agili (II): Scrum da zero
 - 10.9.1. Scrum e la sua applicabilità
 - 10.9.2. Ruoli chiave in Scrum: Product Owner, Scrum Master e Development Team
 - 10.9.3. Artefatti in Scrum: Product Backlog, Sprint Backlog e Incremento
 - 10.9.4. Eventi Scrum: Sprint Planning, Daily Scrum, Sprint Review e Retrospective
- 10.10. Metodologie agili (III): Kanban e Metriche da zero
 - 10.10.1. Kanban e il suo approccio visivo
 - 10.10.2. Elementi chiave in Kanban: Colonne, carte e limiti WIP
 - 10.10.3. Metriche agili: Burnup, burndown charts, velocity e lead time





“

Ti preparerai comodamente da casa tua e al tuo ritmo, con l'innovativa metodologia online e la flessibilità di cui hai bisogno. Non esitare a far decollare la tua carriera tecnologica! Iscriviti subito”

04

Obiettivi didattici

Questo Master ha come obiettivo principale dotare i professionisti delle competenze necessarie per iniziare la loro carriera nel mondo della programmazione in modo solido ed efficace. Durante il programma, svilupperanno una comprensione approfondita dei concetti fondamentali della programmazione, come algoritmi, strutture di dati e logica di programmazione, che costituiscono la base su cui costruiranno competenze più avanzate. Inoltre, acquisiranno una padronanza dei linguaggi di programmazione ampiamente utilizzati nel settore (*Python, JavaScript e HTML*), che consentirà loro di affrontare con fiducia vari progetti tecnologici.



“

Acquisirai le competenze necessarie
per guidare progetti di sviluppo, gestire
database e ottimizzare i processi,
aumentando la tua competitività
professionale nel settore tecnologico”



Obiettivi generali

- Gestire i fondamenti di programmazione e sviluppo del software attraverso l'identificazione degli elementi essenziali di un programma, le strutture di controllo e il rapporto tra software e hardware
- Comprendere i fondamenti di Python e la sua sintassi di base, sviluppando un ambiente adatto per lo sviluppo backend
- Acquisire una solida conoscenza degli algoritmi e delle strutture dei dati fondamentali
- Padroneggiare i fondamenti dello sviluppo backend e il suo ruolo all'interno di un'architettura di software
- Gestire i principi della programmazione orientata agli oggetti e applicarli nella costruzione di software flessibile, riutilizzabile e mantenibile, utilizzando Python come linguaggio di programmazione
- Sviluppare i concetti fondamentali di HTML e CSS per strutturare e dare stile alle pagine web
- Analizzare il linguaggio JavaScript, dai fondamenti alle tecniche avanzate
- Introdurre i concetti fondamentali di React.js per costruire applicazioni web interattive
- Gestire i tipi di database e i relativi linguaggi di query, sia in ambienti relazionali che NoSQL
- Dettagliare l'automazione dei test, dell'integrazione e della distribuzione del codice





Obiettivi specifici

Modulo 1. Programmazione e Sviluppo di Software da Zero

- ◆ Definire e differenziare tra software e hardware
- ◆ Fondare i concetti essenziali della programmazione
- ◆ Conoscere la struttura di base di un programma
- ◆ Esplorare e analizzare i diversi paradigmi di programmazione

Modulo 2. Sviluppo Backend I: Python da Zero

- ◆ Padroneggiare le funzionalità di Python
- ◆ Comprendere la struttura e la sintassi di base di Python
- ◆ Sviluppare competenze nel controllo del flusso tramite condizionali
- ◆ Applicare i loop per creare cicli di ripetizione in Python

Modulo 3. Sviluppo Backend II - Algoritmi e Strutture di Dati con Python da Zero

- ◆ Implementare e confrontare i tipi di algoritmi di ricerca nelle strutture dati
- ◆ Analizzare algoritmi di ordinamento come bolla, inserimento, selezione, merge sort e quick sort
- ◆ Esaminare la complessità algoritmica e la misurazione dell'efficienza utilizzando la notazione Big O
- ◆ Rappresentare i grafi ed eseguire percorsi in profondità (DFS) e in larghezza (BFS)

Modulo 4. Sviluppo Backend III - Flask, Creazione di API e Architettura di Base da Zero

- Sviluppare API RESTful utilizzando Flask
- Integrare database in applicazioni Flask
- Implementare autenticazione e sicurezza nelle API
- Progettare l'architettura di base di un'applicazione backend con Flask

Modulo 5. Programmazione Orientata a Oggetti e Modelli di Progettazione da Zero

- Definire i concetti chiave della programmazione orientata agli oggetti come classi, oggetti, attributi, metodi, encapsulamento, astrazione, ereditarietà e polimorfismo
- Comprendere l'uso di encapsulamento e astrazione in classi utilizzando il linguaggio di programmazione Python
- Esaminare il concetto di polimorfismo e sovraccarico all'interno del linguaggio Python comprendendo le sue applicazioni e i vantaggi
- Determinare i tipi di relazioni tra le classi come associazione, aggregazione e composizione

Modulo 6. Frontend I - HTML e CSS da Zero

- Identificare la struttura di base di un documento HTML e la sua importanza nello sviluppo web
- Utilizzare HTML per organizzare e presentare il contenuto sul web in modo semantico e accessibile: Architettura web
- Applicare stili con CSS per migliorare la presentazione visiva degli elementi
- Utilizzare il modello box di CSS per strutturare e distribuire gli elementi nell'interfaccia



Modulo 7. Frontend II – JavaScript da Zero

- ◆ Comprendere la sintassi e i tipi di dati in JavaScript
- ◆ Imparare a strutturare il codice utilizzando funzioni e strutture di controllo
- ◆ Manipolare il DOM per interagire con le pagine web dinamiche
- ◆ Lavorare con le API e gestire la sincronizzazione tramite promesse e `async/await`

Modulo 8. Frontend III - React.js da Zero

- ◆ Comprendere l'uso di JSX per creare interfacce dichiarative
- ◆ Imparare a lavorare con componenti funzionali, `prop` e cicli di vita
- ◆ Gestire gli stati locali e globali utilizzando strumenti moderni come Context API e Redux Toolkit
- ◆ Implementare il routing per costruire applicazioni a pagina singola (SPA)

Modulo 9. Gestione e Ottimizzazione dei Database da Zero

- ◆ Riconoscere i tipi di database e le loro caratteristiche
- ◆ Comprendere e applicare il modello relazionale dei dati
- ◆ Sviluppare competenze in SQL per la gestione di database
- ◆ Utilizzare query avanzate in SQL

Modulo 10. Strumenti di Sviluppo da Zero: Linux, Controllo di Versione, CI/CD, Docker e Metodologie Agili

- ◆ Gestire il sistema operativo Linux a livello di riga di comando
- ◆ Padroneggiare l'uso di Git per il controllo delle versioni
- ◆ Implementare pipeline di integrazione e distribuzione continua (CI/CD)
- ◆ Creare e gestire i container Docker

“

Vuoi padroneggiare il linguaggio del futuro? Questo Master rappresenta il primo passo verso il futuro tecnologico che hai sempre desiderato. Iscriviti subito!

05

Opportunità professionali

Questa qualifica dà accesso ad una vasta gamma di opportunità professionali in un settore tecnologico in continua espansione. Man mano che le aziende di tutti i settori si digitalizzano, la domanda di professionisti con competenze di programmazione non solo aumenta, ma si diversifica, aprendo opportunità in settori diversi come lo sviluppo di software, la gestione di database, web design e automazione dei processi. Al termine, gli studenti saranno perfettamente qualificati per assumere ruoli in aziende tecnologiche, *startup* innovative e organizzazioni di qualsiasi settore che cercano di integrare soluzioni digitali nelle loro operazioni.





“

Non solo diventerai un esperto di alto livello, ma affronterai anche una vasta gamma di opportunità professionali, assicurando il tuo successo in un mercato del lavoro in continua crescita e trasformazione”

Profilo dello studente

Lo studente sarà un professionista altamente qualificato per affrontare le sfide tecnologiche del futuro. In questo senso, non solo padroneggerà i concetti fondamentali di algoritmi, strutture dati e linguaggi di programmazione, ma avrà anche sviluppato una profonda comprensione degli strumenti e delle tecnologie più richieste nel settore. Con una solida base nella programmazione e nello sviluppo di applicazioni, questo esperto sarà in grado di affrontare i progetti tecnologici in modo autonomo ed efficace. Inoltre, si distinguerà per la capacità di risolvere problemi complessi, applicare soluzioni innovative e gestire sistemi e database.

Diventerai un professionista versatile, con una solida preparazione tecnica e una visione strategica, in grado di integrare la programmazione in soluzioni tecnologiche che rispondono alle imprese e alla società.

- ◆ **Pensiero critico e problem solving:** Analizzare situazioni complesse e trovare soluzioni innovative ed efficienti attraverso la programmazione
- ◆ **Lavoro di squadra e collaborazione multidisciplinare:** Lavorare in team multidisciplinari, collaborando efficacemente con professionisti di diversi settori, come progettisti, ingegneri e analisti di dati, per raggiungere obiettivi comuni nei progetti tecnologici
- ◆ **Gestione del tempo e autonomia nell'apprendimento:** Gestire i tempi in modo efficace per affrontare scadenze strette e sviluppare progetti in modo autonomo e responsabile
- ◆ **Adattamento alle nuove tecnologie:** Gestire nuovi strumenti e linguaggi di programmazione, assicurandone il costante aggiornamento in un ambiente tecnologico in rapida evoluzione



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

- 1. Sviluppo di Software:** Responsabile di progettazione, creazione e manutenzione di applicazioni e software in base alle esigenze del cliente o dell'azienda.
- 2. Sviluppatore Web:** Gestione di creazione, progettazione e manutenzione di siti web, garantendo la loro funzionalità e ottimizzazione su diversi dispositivi.
- 3. Ingegnere di Database:** Responsabile di progettazione, implementazione e gestione di database, garantendone efficienza, sicurezza e disponibilità.
- 4. Analista di Sistemi:** Sviluppatore di sistemi informatici esistenti, proponendo miglioramenti per ottimizzare le prestazioni e l'efficienza dei processi.
- 5. Programmatore Front-End:** Responsabile dello sviluppo della parte visiva di applicazioni e siti web, migliorando l'esperienza dell'utente.
- 6. Programmatore Back-End:** Responsabile per lo sviluppo della logica del server e dei database di applicazioni e siti web, garantendone il corretto funzionamento.
- 7. Amministratore di Reti:** Responsabile della gestione e della manutenzione delle infrastrutture di rete all'interno di un'azienda, garantendone la stabilità e la sicurezza.
- 8. Consulente Tecnologico:** Guida e consulenza nelle aziende, fornendo informazioni sull'implementazione di soluzioni tecnologiche adeguate alle loro esigenze, migliorando le loro prestazioni e competitività.
- 9. Sviluppo di Applicazioni Mobili:** Responsabile di progettazione, sviluppo e manutenzione di applicazioni per dispositivi mobili, garantendone l'usabilità e le prestazioni.
- 10. Ingegnere di Software di Automazione:** Responsabile della progettazione e dello sviluppo di soluzioni automatizzate per migliorare l'efficienza dei processi aziendali, utilizzando la programmazione e la tecnologia avanzata.

06

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 ripetere 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, coinvolgendi 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

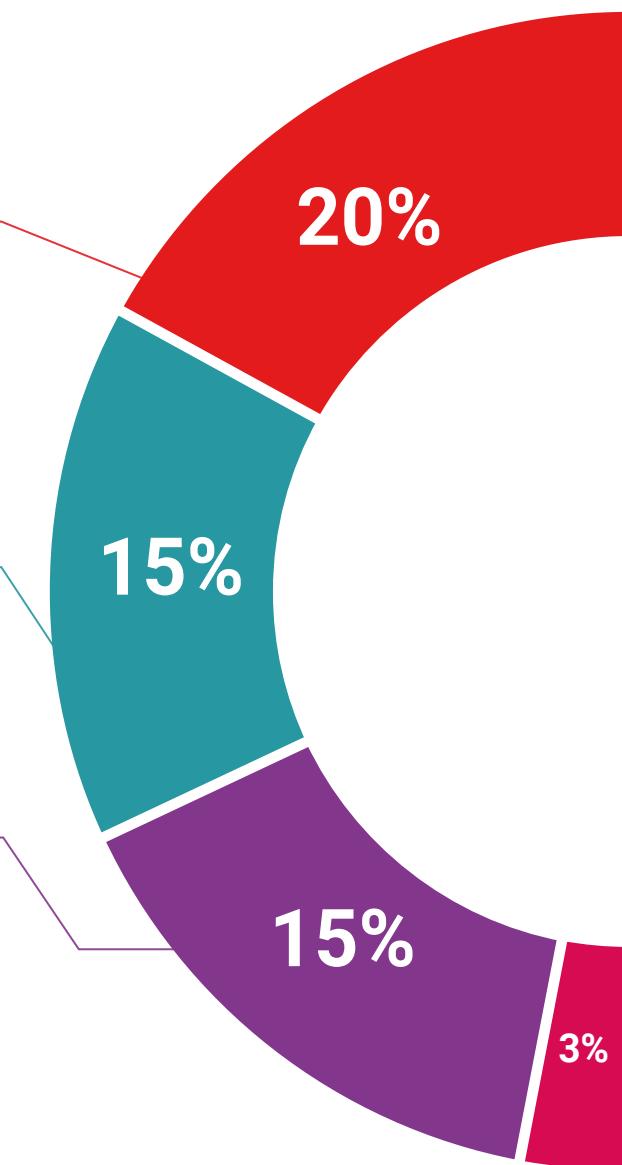
Presentiamo i contenuti in modo accattivante e dinamico tramite strumenti multimediali che includono audio, video, immagini, diagrammi e mappe concettuali per consolidare la conoscenza.

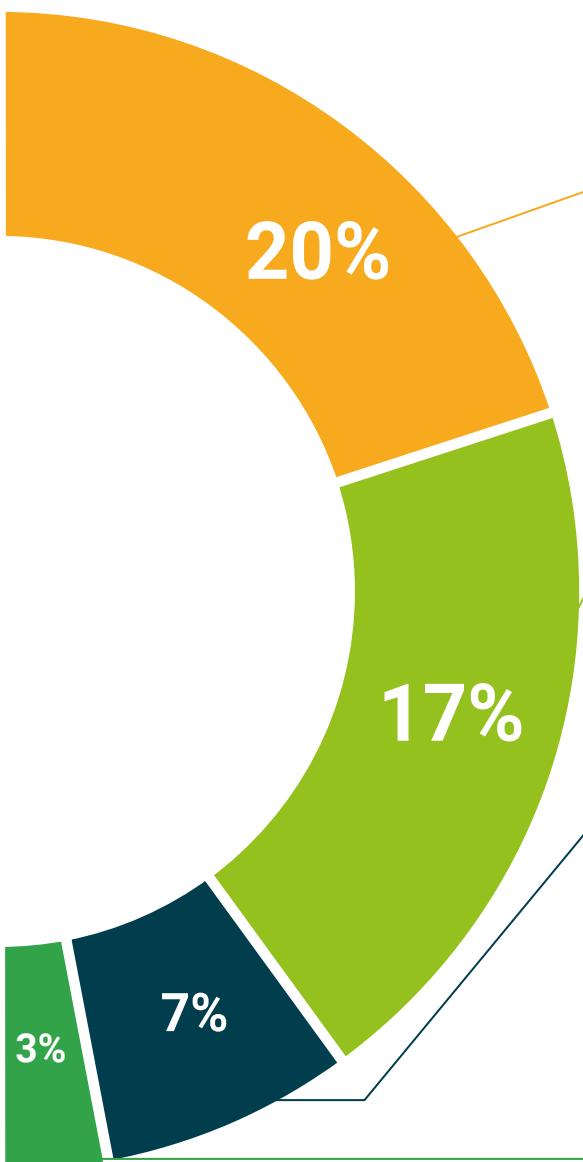
Questo esclusivo sistema di preparazione per la presentazione di contenuti multimediali è stato premiato da Microsoft come "Caso di successo in Europa".



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 è composto da professionisti altamente qualificati e con una vasta esperienza nel settore tecnologico. Ognuno è stato accuratamente selezionato per garantire che gli studenti ricevano una formazione di qualità, basata su conoscenze aggiornate e sulla pratica reale del settore. In questo senso, la loro esperienza di lavoro in aziende rinomate e la preparazione accademica consentono di offrire una prospettiva globale e applicata sulla Programmazione. In definitiva, i mentori sono esperti in diversi settori della tecnologia, fornendo una visione multidisciplinare e arricchente.



“

Il personale docente è un pilastro fondamentale per il successo del programma. La qualità e l'esperienza dei mentori garantiranno una formazione completa che ti preparerà ad affrontare il settore tecnologico con fiducia e competenza”

Direzione**Dott. Lucas Cuesta, Juan Manuel**

- Ingegnere Software Senior e Analista presso Indizen - Believe in Talent
- Ingegnere Software Senior e Analista presso Krell Consulting e IMAGiNA Artificial Intelligence
- Ingegnere Software presso Intel Corporation
- Ingegnere software presso Intelligent Dialogue Systems
- Dottorato in Ingegneria dei Sistemi Elettronici per Ambienti Intelligenti presso l'Università Politecnica di Madrid
- Laureato in Ingegneria delle Telecomunicazioni presso l'Università Politecnica di Madrid
- Master in Ingegneria dei Sistemi Elettronici per Ambienti Intelligenti presso l'Università Politecnica di Madrid

**Dott. Márquez Ruiz de Lacanal, Juan Antonio**

- Sviluppatore di Software presso GTD Defense & Security Solutions
- Sviluppatore di Software presso Solera Inc
- Ingegnere di Sviluppo e Ricerche presso GRVC Siviglia
- Cofondatore di Unmute
- Cofondatore di VR Educa
- Scambio accademico in Ingegneria e Imprenditorialità presso la Berkeley University of California
- Laurea in Ingegneria Industriale presso l'Università di Siviglia

Personale docente

Dott. Luna Perejón, Francisco

- ◆ Specialist in Architettura e Tecnologia dei Computer
- ◆ Dottorato in Ingegneria Informatica presso l'Università di Siviglia
- ◆ Master Universitario in Ingegneria Informatica presso l'Università di Siviglia
- ◆ Laurea in Ingegneria della Salute presso l'Università di Siviglia
- ◆ Laurea in Ingegneria e Tecnologie Informatiche presso l'Università di Siviglia
- ◆ Membro di: Gruppo di ricerca in Robotica e Tecnologie Informatiche (TEP108)

Dott. Péris Millán, Eduardo

- ◆ Direttore dell'Area Tecnologica di Consulenza
- ◆ Specialist in Ingegneria Informatica
- ◆ Master in Gestione strategica delle informazioni e della conoscenza nelle organizzazioni
- ◆ Master in Leadership e Gestione Pubblica
- ◆ Esperto di Gestione Pubblica
- ◆ Esperto di Sistemi Informatici per Smart Cities

Dott. Pi Morell, Oriol

- ◆ Analista Funzionale presso Fihoca
- ◆ Product Owner di Hosting e posta elettronica in CDmon
- ◆ Analista Funzionale e Software Engineer in Atmira e CapGemini
- ◆ Docente in Capgemini, Forma Capgemini e Atmira
- ◆ Laurea in Ingegneria Tecnica dell'Informatica di Gestione presso l'Università Autonoma di Barcellona
- ◆ Master in Intelligenza Artificiale presso l'Università Cattolica di Ávila
- ◆ MBA in Gestione e Amministrazione d'Impresa presso IMF Smart Education
- ◆ Master in Gestione dei Sistemi di Informazione con IMF Smart Education
- ◆ Corso di Specializzazione in Design Patterns dell'Universitat Oberta de Catalunya (UOC)

Dott. Grillo Hernández, José Enrique

- ◆ Sviluppatore di applicazioni e analista tecnologico
- ◆ Sviluppatore Senior di Applicazioni Mobili presso Globant
- ◆ Sviluppatore Android presso Plexus Tech
- ◆ Sviluppatore Senior Android presso RoadStr
- ◆ Sviluppatore Senior Mobile presso Avantgarde IT-Information Technology Services
- ◆ Project Leader presso Smartdess
- ◆ Sviluppatore presso Educatablet
- ◆ Analista della Tecnologia presso Corporate Mobile Solutions
- ◆ Master in Ingegneria dei Sistemi presso l'Università Simón Bolívar

Dott.ssa Domínguez Valderrama, Desirée

- ◆ Lead Product & Growth Strategist
- ◆ Master in Graphic Design e Creatività presso la Business School della Camera di Commercio di Siviglia
- ◆ Esperta in UX/UI Designer presso CoderHouse
- ◆ Esperta in Tecnologia e Imprenditorialità



Cogli l'occasione per conoscere gli ultimi sviluppi in questo campo e applicarli alla tua pratica quotidiana"

08

Titolo

Il Master in Programmazione da Zero garantisce, oltre alla preparazione più rigorosa e aggiornata, il conseguimento di una qualifica di Master rilasciata da TECH Global University.



66

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

Questo programma ti consentirà di ottenere il titolo di studio privato di **Master in Programmazione da Zero** 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 Programmazione da Zero**

Modalità: **online**

Durata: **12 mesi**

Accreditamento: **60 ECTS**



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
Programmazione
da Zero

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

Master

Programmazione da Zero

