

Master Privato

Programmazione Front
End Full Stack Developer



Master Privato Programmazione Front End Full Stack Developer

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

Accesso al sito web: www.techtute.com/it/informatica/master/master-programmazione-front-end-full-stack-developer

Indice

01

Presentazione

pag. 4

02

Obiettivi

pag. 8

03

Competenze

pag. 14

04

Direzione del corso

pag. 18

05

Struttura e contenuti

pag. 22

06

Metodologia

pag. 34

07

Titolo

pag. 42

01

Presentazione

I *Front End* Developer sono una parte indispensabile di qualsiasi sviluppo web. La loro natura di anello di congiunzione tra la progettazione grafica e la programmazione *Back End* richiede conoscenze molto specifiche e sviluppate, che a loro volta devono essere aggiornate continuamente per rispondere alle sfide sempre più impegnative del presente. Essendo una specializzazione che offre molteplici opportunità a livello professionale, TECH ha riunito il miglior personale docente possibile per sviluppare un programma universitario ampio, completo e metodologico. Incentrato sugli strumenti e sulle metodologie più avanzate nello sviluppo di soluzioni web, questo corso fornisce l'impulso necessario per fare un passo avanti nella programmazione *Front End Full Stack*.



“

Diventa un elemento essenziale di qualsiasi sviluppo web, in grado di guidare e coordinare team informatici grazie alla metodologia di lavoro e di leadership perfezionata da questo Master Privato"

Lo sviluppo *Full Stack* è un'opzione particolarmente interessante per tutti i professionisti informatici che desiderino dare un impulso significativo alla propria carriera. Le competenze richieste per operare al meglio nel settore sono ampie, il che significa che le opportunità di crescita e anche di guidare team di sviluppatori sono molteplici.

Grazie all'esaustività con cui sono stati sviluppati i contenuti di questo programma, lo studente sarà in grado di indirizzare la propria carriera verso lo sviluppo web *Front End*, l'impaginazione, la customer experience o il DevOps. Con una visione a 360° dell'intero processo di realizzazione di un'applicazione/web, l'informatico sarà in grado di affrontare qualsiasi tipo di progetto, incorporando inoltre i più recenti progressi legati al ciclo di vita del software.

Un'opportunità accademica unica per avere accesso a conoscenze che combinano la più recente teoria informatica con una pratica professionale di alto livello, fornita da un personale docente di altissima qualità. La sua esperienza a capo di numerosi progetti rilevanti nel campo del digital banking o delle telecomunicazioni rende più ricco il contenuto didattico, fornendo un gran numero di casi reali e di letture complementari.

La grande flessibilità di questo insegnamento è un'altra delle sue caratteristiche più rilevanti. Non ci sono orari fissi o lezioni frontali, e lo studente decide quando, dove e come seguire l'intero carico didattico. Tutti i contenuti sono disponibili fin dal primo giorno del corso, e possono essere studiati da qualsiasi dispositivo dotato di connessione a internet.

Questo **Master Privato in Programmazione Front End Full Stack Developer** possiede il programma più completo e aggiornato del mercato. Le caratteristiche principali del programma sono:

- ◆ Sviluppo di casi pratici presentati da esperti in Programmazione Front-end Full Stack Developer
- ◆ 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
- ◆ La sua speciale enfasi sulle metodologie innovative
- ◆ Lezioni teoriche, domande all'esperto e lavori di riflessione individuale
- ◆ Contenuti disponibili da qualsiasi dispositivo fisso o portatile provvisto di connessione a internet



Iscriviti subito e non perdere l'opportunità che ti farà arrivare sulla cima nella leadership e nello sviluppo dei progetti informatici più ambiziosi"

“

Raggiungerai un livello avanzato di competenza, essendo in grado di proporre qualsiasi soluzione web richiesta da una prospettiva di Customer Experience moderna e adeguata al mercato attuale"

Il personale docente del programma comprende rinomati professionisti del settore nonché riconosciuti specialisti appartenenti a società scientifiche e università prestigiose, 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 rinomata fama.

Approfondirai lo studio delle metodologie agili e di come possono essere implementate nel processo di sviluppo, aumentando le tue capacità e competenze trasversali.

Avrai accesso a un'ampia gamma di materiali didattici, che include il linguaggio di programmazione Javascript a strumenti come CSS, Angular e ReactJS.



02

Obiettivi

L'obiettivo di questo Master Privato in Programmazione Front End Full Stack Developer, tenendo conto delle molteplici opzioni che lo sviluppo offre, è quello di fornire le conoscenze e le tecniche più avanzate in questo campo. Gli informatici potranno quindi iniziare a sviluppare i propri progetti o a dare impulso alla propria carriera professionale ancor prima di aver completato il corso, grazie all'approccio eminentemente pratico di tutti i contenuti proposti.



“

Potrai raggiungere i tuoi obiettivi di carriera più ambiziosi grazie all'approccio distintivo di questo programma, che ti condurrà attraverso le diverse fasi di sviluppo front-end full stack"



Obiettivi generali

- ◆ Generare conoscenze specialistiche sugli aspetti chiave della programmazione
- ◆ Promuovere il pensiero da un punto di vista algoritmico
- ◆ Fornire gli strumenti e le competenze necessarie per lo sviluppo
- ◆ Promuovere l'adozione di metodologie agili per l'attuazione dei progetti
- ◆ Sviluppare una conoscenza specialistica sui fondamenti della web
- ◆ Incoraggiare l'uso di strumenti e tecniche di sviluppo *Front End* moderni
- ◆ Sviluppare un web design per affrontare il layout in modo corretto
- ◆ Valutare le conoscenze acquisite

“

Migliorerai le tue capacità e competenze in modo progressivo, nel corso di 10 moduli elaborati a partire dalle conoscenze più solide e dalla comprovata esperienza dei docenti”





Obiettivi specifici

Modulo 1. Sviluppo *Full stack developer*

- ◆ Sviluppare competenze avanzate di programmazione
- ◆ Incoraggiare l'uso di sistemi di controllo di versioni e piattaforme di hosting del codice
- ◆ Promuovere l'uso della metodologia agile
- ◆ Approfondire i concetti chiave e il funzionamento di Internet
- ◆ Aumentare le competenze nell'uso della riga di comando

Modulo 2. *Front-end* nella programmazione

- ◆ Identificare e comprendere la sintassi corretta di HTML e CSS
- ◆ Esplorare i diversi elementi dell'HTML
- ◆ Determinare l'approccio della progettazione adattativa
- ◆ Utilizzare la formattazione di presentazione per le pagine web applicando i fogli di stile a cascata
- ◆ Incorporare il preprocessore CSS
- ◆ Stabilire i vantaggi dell'uso di un preprocessore
- ◆ Generare conoscenze specialistiche sui sistemi di progettazione
- ◆ Stabilire i criteri per l'utilizzo del sistema di progettazione

Modulo 3. Linguaggio JAVASCRIPT applicato al *Full Stack Developer*

- ◆ Stabilire i tipi di base e complessi offerti da JavaScript
- ◆ Analizzare i diversi modi di programmare con questo linguaggio e come farne un uso corretto nelle diverse situazioni
- ◆ Aggiornare le conoscenze sulle ultime versioni
- ◆ Comprendere la programmazione funzionale
- ◆ Esaminare la programmazione asincrona e le sue caratteristiche

Modulo 4. Layout web applicato al *Full Stack Developer*

- ◆ Valutare un web design per sapere come posizionarlo temporaneamente
- ◆ Esaminare le principali regole CSS
- ◆ Presentare diverse metodologie CSS per ottenere progetti *responsive*
- ◆ Conoscere i fondamenti dei principi di sviluppo a cascata CSS
- ◆ Identificare la tecnologia Bootstrap in qualsiasi progetto web
- ◆ Analizzare i principi di Bootstrap
- ◆ Sviluppare un mockup web utilizzando Bootstrap
- ◆ Determinare i principi di sviluppo in un progetto SaSS

Modulo 5. Strumenti JavaScript. Libreria ReactJs

- ◆ Determinare le funzioni di React
- ◆ Impostare un progetto con Create React App
- ◆ Analisi del ciclo di vita dei componenti in React
- ◆ Generare conoscenze specialistiche sulle moderne funzionalità di React come *Hooks* e *Context*
- ◆ Impostare gli stati globali utilizzando *Context*
- ◆ Creazione e rendering di elenchi e creazione di moduli con React
- ◆ Implementare la convalida dei campi nei moduli
- ◆ Styling di componenti ed elementi
- ◆ Debug, test e deployment di applicazioni React

Modulo 6. Framework di JavaScript. Angular

- ◆ Sviluppare conoscenze specialistiche sull'architettura del *Framework*
- ◆ Approfondire la metodologia di Angular
- ◆ Analizzare il concetto di componenti
- ◆ Organizzare correttamente il codice

Modulo 7. Programmazione in linguaggio NodeJS

- ◆ Generare conoscenze specializzate sui tipi di JavaScript e sui suoi operatori
- ◆ Analizzare i modi migliori per programmare con il linguaggio
- ◆ Aggiornare le conoscenze sulle ultime versioni
- ◆ Esplorare la programmazione funzionale
- ◆ Sviluppare la programmazione asincrona e la sua giustificazione
- ◆ Acquisire la capacità di realizzare un'applicazione con NodeJSINDEX





Modulo 8. Database per sviluppatori *Full Stack*

- ◆ Determinare perché utilizzare un database nello sviluppo di un'applicazione
- ◆ Esaminare i tipi di database disponibili e le loro differenze
- ◆ Sviluppare un'idea chiara dell'utilizzo di ciascun tipo di database
- ◆ Analizzare l'uso dei database negli attuali paradigmi di sviluppo

Modulo 9. UX CX. Esperienza del cliente

- ◆ Analizzare l'importanza dell'utente oggi e approfondire la cultura del feedback
- ◆ Realizzare strategie omnichannel e di personalizzazione basate su microinterazioni
- ◆ Studiare l'evoluzione della web analytics fino all'analisi comportamentale
- ◆ Scoprite come l'intelligenza artificiale ha portato la CX a un livello superiore
- ◆ Stabilire le più importanti tecniche di analisi dell'esperienza web, della mobilità e dell'accessibilità
- ◆ Presentare la metodologia del *Design Thinking* e il processo di creazione dell'esperienza utente
- ◆ Presentare strumenti concreti di prototipazione e *WireFraming*, così come *Frameworks* di sviluppo di *Front End*

Modulo 10. Integrazione continua e deployment delle applicazioni

- ◆ Analizzare i vantaggi dell'adozione di un modello di distribuzione automatica delle applicazioni
- ◆ Stabilire le differenze tra continuous integration, continuous delivery e continuous deployment
- ◆ Determinare le caratteristiche principali di DevOps
- ◆ Valutare alcuni degli strumenti principali per l'implementazione di pipeline CI/CD
- ◆ Conoscere i fattori essenziali per lo sviluppo di applicazioni preparate a supportare i processi CI/CD
- ◆ Esaminare le tecnologie dei container come pilastro fondamentale della pratica CI/CD

03

Competenze

Le competenze che un programmatore *Front End Full Stack* deve sviluppare sono molteplici, soprattutto in un campo altamente competitivo e specializzato come quello dell'informatica. Il programma copre pertanto i diversi tipi di linguaggio più comuni in questo campo, nonché gli strumenti e la filosofia di lavoro che l'informatico deve seguire per fare la differenza nella sua professione. Questo è possibile grazie alla natura multidisciplinare del personale docente stesso, che ha messo a disposizione le sue conoscenze relative alle diverse aree di sviluppo *Front End*.



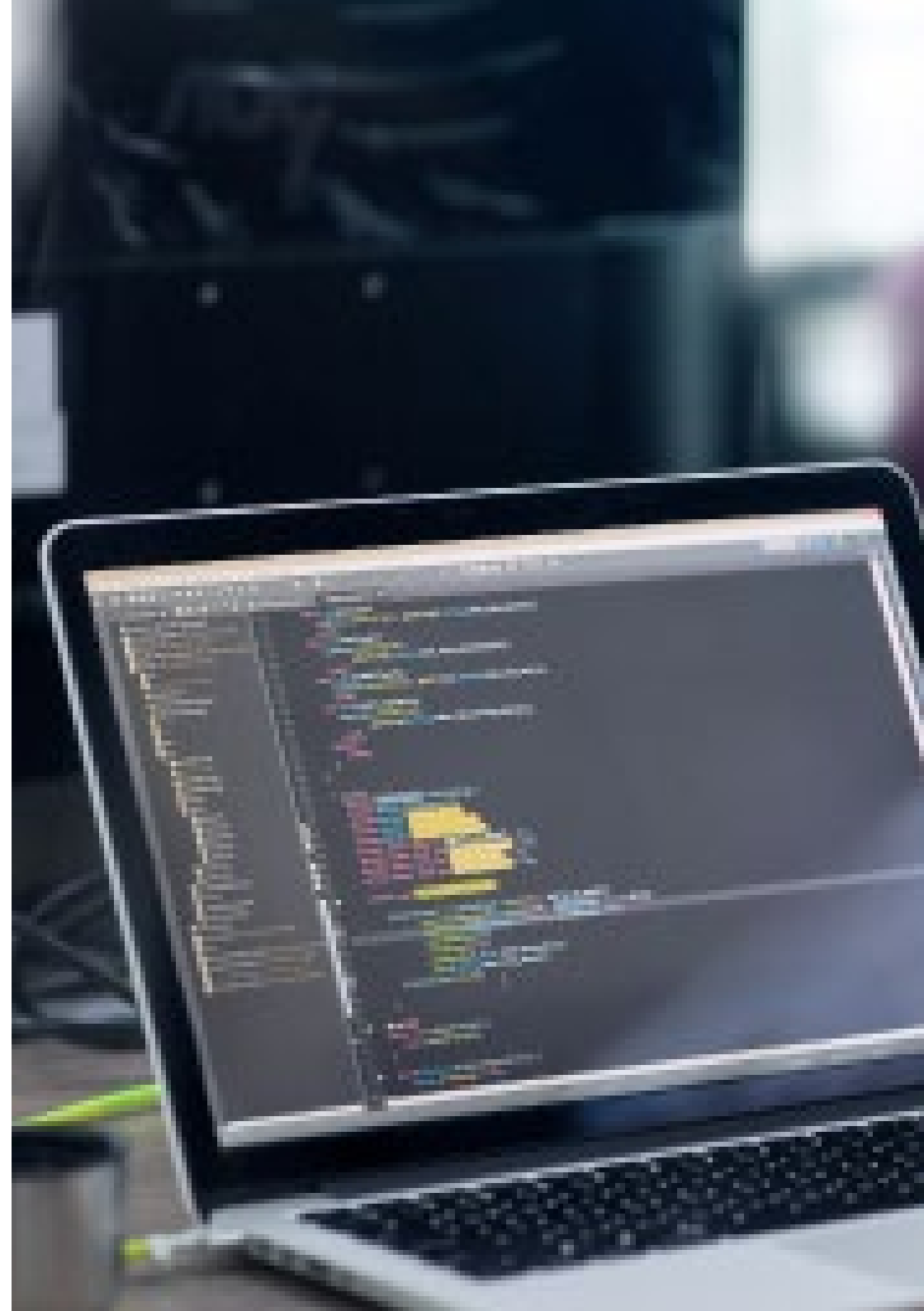
“

Amplia il tuo CV grazie a una vasta gamma di abilità e competenze di alta qualità richieste per lo sviluppo software"



Competenze generali

- ◆ Riconoscere correttamente la sintassi dei linguaggi HTML e CSS
- ◆ Sviluppare criteri di best practice per lo sviluppo web
- ◆ Generare conoscenze specialistiche sul linguaggio JavaScript
- ◆ Essere in grado di sviluppare qualsiasi tipo di applicazione con JavaScript
- ◆ Analizzare la libreria Bootstrap
- ◆ Realizzare progetti di impaginazione con SaSS (*Syntactically Awesome Stylesheets*)
- ◆ Identificare la sintassi di React e come utilizzarla per programmare
- ◆ Applicare le migliori pratiche al linguaggio
- ◆ Esaminare il processo di caricamento e accesso di ciascuno dei principali tipi di database della rispettiva area
- ◆ Valutare gli strumenti e le tecniche più importanti per l'analisi della CX e dello "stack tecnologico" comunemente utilizzato nelle aziende





Competenze specifiche

- ◆ Analizzare le diverse strutture di dati
- ◆ Esaminare le tecniche di progettazione e interpretazione degli algoritmi
- ◆ Preparazione degli ambienti di sviluppo
- ◆ Clonare un sito web
- ◆ Costruire un sito web con Bootstrap
- ◆ Compilare il codice CSS con SaSS
- ◆ Sviluppare un *Framework* CSS basato su Bootstrap utilizzando SaSS
- ◆ Generare un progetto e metterlo in pratica
- ◆ Stabilire come connettersi e caricare/estrarre dati da diversi tipi di database
- ◆ Identificare pratiche, casi d'uso, tecnologie e strumenti dell'ecosistema CI/CD, essenziali per supportare l'intero processo



Nel corso del corso svilupperai le competenze necessarie per avere successo nello sviluppo Front End, migliorando non solo le tue conoscenze ma anche le tue competenze trasversali"

04

Direzione del corso

TECH pone particolare attenzione alla selezione del personale docente responsabile dei suoi corsi. L'accuratezza con cui le squadre sono state messe insieme garantisce non solo la qualità dei contenuti in sé, ma anche il coinvolgimento di un personale docente impegnato nel miglioramento professionale degli informatici. Lo studente avrà quindi a disposizione un materiale didattico adeguato agli standard più esigenti, con il supporto e il materiale complementare che solo può fornire la più grande istituzione accademica online del mondo.



“

In caso di dubbi potrai quindi interpellare direttamente il personale docente, ottenendo un tutoraggio personalizzato e adeguato alle tue esigenze"

Direzione



Dott. Olalla Bonal, Martín

- ◆ Specialista tecnico *Blockchain* presso IBM SPGI
- ◆ Tecnico commerciale specializzato in *Blockchain*. IBM
- ◆ Direttore di Architettura. *Blocknitive*
- ◆ Tecnico di Elettronica Digitale
- ◆ *Blockchain Architect* -IT *Infrastructure Architect* - Gestore di progetti IT. Aree di attività: Software, Infrastruttura, Telecomunicazioni

Personale docente

Dott. Calzada Martínez, Jesús

- ◆ Ingegnere software senior presso Devo
- ◆ Sviluppatore *Full Stack* presso Blocknitive
- ◆ Manager *Front End* presso Infinia
- ◆ Sviluppatore *Full Stack* presso Resem
- ◆ Sviluppatore Java presso Hitec
- ◆ Laurea in Ingegneria dei computer

Dott. Guerrero Díaz-Pintado, Arturo

- ◆ Technical Pre-Sales Engineer mediante Watson *Customer Engagement portfolio* (soluzioni di Marketing e *Customer Experience*) in Spagna, Portogallo, Grecia e Israele presso IBM
- ◆ Ingegnere di rete R&D presso Telefónica
- ◆ Consulente di servizi professionali per organizzazioni all'avanguardia in Europa, Medio Oriente e America Latina da IBM.
- ◆ Laurea in Ingegneria delle Telecomunicazioni presso l'Università di Alcalá e la *Danish Technical University*
- ◆ Collaborazioni eccezionali con rinomate università e centri di istruzione superiore in materie legate alla tecnologia come Intelligenza Artificiale, *Internet of Things*, *Cloud*, *Customer Experience* y *Digital Transformation*

Dott. Pintado San Claudio, Bruno

- ◆ Coordinatore dello sviluppo presso IDavinci
- ◆ Sviluppatore Java presso la Biblioteca Nazionale di Spagna
- ◆ Sviluppatore di supporto e tecnico di rete N1 presso Sanitas
- ◆ Tecnico di supporto ai sistemi presso il Comune di Alcobendas
- ◆ Tecnico delle comunicazioni N1 per l'ADIF presso il Centro di telecomunicazioni di Atocha
- ◆ Laurea in Ingegneria Tecnica delle Telecomunicazioni, con specializzazione in Sistemi Elettronici, presso l'Università Politecnica di Madrid
- ◆ Laurea in Ingegneria Elettronica delle Comunicazioni presso l'Università Politecnica di Madrid

Dott. Reyes Oliva, Luis

- ◆ Sviluppatore e architetto cloud presso IBM
- ◆ Technical Client Manager per i conti integrati di BBVA presso IBM
- ◆ *Cloud Executive Selling* presso IBM
- ◆ Architetto Cloud e DevOps presso IBM
- ◆ Architetto software clienti presso Telefónica
- ◆ Architetto di soluzioni tecniche presso Rational
- ◆ *Software Engineering Manager* presso Borland
- ◆ Responsabile sviluppo software e assicurazione qualità presso Altana Consulting
- ◆ Laurea in Ingegneria Informatica presso l'Università Pontificia di Salamanca, Madrid

Dott. Frias Favero, Pedro Luis

- ◆ CTO- Swearit Technologies
- ◆ COO - Key Identification
- ◆ Esperto in *blockchain* e applicazioni decentralizzate - Universidad de Alcalá
- ◆ Full Stack Developer - Ironhack
- ◆ Ingegnere industriale presso l'Università di Yacambu

Dott. Gómez Rodríguez, Antonio

- ◆ Ingegnere di soluzioni cloud presso Oracle
- ◆ Project Manager presso Sopra Group
- ◆ Project Manager presso Everis
- ◆ Project Manager nell'Azienda Pubblica di Gestione dei Programmi Culturali Consiglio della Cultura dell'Andalusia
- ◆ Analista di sistemi informativi. Sopra Group
- ◆ Laurea in Ingegneria delle Telecomunicazioni presso l'Università Politecnica della Catalogna
- ◆ Post-laurea in Tecnologie e Sistemi Informatici, Istituto Catalano di Tecnologia
- ◆ *E-Business Master*, Business School La Salle

05

Struttura e contenuti

Considerando lo sforzo e l'investimento che l'informatico deve compiere per svolgere un programma di questa natura, TECH offre una metodologia didattica di prim'ordine, comprovata in termini di efficienza e rendimento. Il *Relearning* permette allo studente di acquisire le conoscenze più importanti durante l'intero programma, evitando così di sprecare tempo prezioso che può dedicare allo studio dei numerosi materiali complementari che accompagnano ogni argomento proposto.



“

Esplora i diversi moduli e argomenti didattici grazie a video riassuntivi, dettagliati e motivazionali creati dai propri docenti”

Modulo 1. Sviluppo *Full stack developer*

- 1.1. Sviluppo *Full Stack Developer* I. Programmazione e linguaggi
 - 1.1.1. Programmazione
 - 1.1.2. Ruoli di programmazione
 - 1.1.3. Linguaggi e *Framework*
 - 1.1.4. Algoritmo
 - 1.1.5. Caratteristiche di un algoritmo
- 1.2. Sviluppo *Full stack developer* II. Tipologia
 - 1.2.1. Variabili e costanti
 - 1.2.2. Tipologie
 - 1.2.3. Operatori
 - 1.2.4. Dichiarazioni
 - 1.2.5. Loop
 - 1.2.6. Funzioni e oggetti
- 1.3. Strutture dati nello sviluppo
 - 1.3.1. Tipi di struttura lineari
 - 1.3.2. Tipi di struttura funzionali
 - 1.3.3. Tipi di strutture ad albero
- 1.4. Progettazione e interpretazione di algoritmi
 - 1.4.1. Parallelismo nello sviluppo. Divide et impera
 - 1.4.2. Algoritmi greedy
 - 1.4.3. Programmazione dinamica
- 1.5. Ambiente e strumenti per lo sviluppo orientato agli sviluppatori al *Full Stack Developer*
 - 1.5.1. Preparazione dell'ambiente per Mac OS
 - 1.5.2. Preparazione dell'ambiente per Linux
 - 1.5.3. Preparazione dell'ambiente per Windows
- 1.6. Linea di comando. Tipologia e funzionamento
 - 1.6.1. Il terminale
 - 1.6.2. Emulatori
 - 1.6.3. Interprete dei comandi
 - 1.6.4. Primi comandi
 - 1.6.5. Navigazione
 - 1.6.6. Gestione di file e cartelle tramite l'interfaccia della riga di comando
 - 1.6.7. *Secure Shell*. SSH
 - 1.6.8. Comandi avanzati
- 1.7. Git. Repository di software
 - 1.7.1. Git repository di software
 - 1.7.2. Uso di Git
 - 1.7.3. Repository di software
 - 1.7.4. Rami
 - 1.7.5. Ciclo di lavoro
 - 1.7.6. Comandi
- 1.8. Servizio di hosting per il versionamento del codice
 - 1.8.1. Servizio di hosting per il versionamento del codice
 - 1.8.2. Fornitori
 - 1.8.3. Repository
- 1.9. Internet
 - 1.9.1. Internet
 - 1.9.2. Protocolli utilizzati nel WWW
 - 1.9.3. Protocollo HTTP
- 1.10. Metodologie di sviluppo *Full Stack*
 - 1.10.1. *Scrum*
 - 1.10.2. XP
 - 1.10.3. *Design Sprint*

Modulo 2. *Front-end* nella programmazione

- 2.1. Linguaggio HTML
 - 2.1.1. Documento HTML
 - 2.1.2. Elemento *Head*
 - 2.1.3. Elemento *Body*
 - 2.1.4. Testo
 - 2.1.5. Hyperlink
 - 2.1.6. Immagini
 - 2.1.7. *Primer Site*
- 2.2. Linguaggio HTML. *Layout*
 - 2.2.1. Linguaggio HTML. Elementi
 - 2.2.2. *Layout* tradizionale
 - 2.2.3. *Layout* semantico
- 2.3. Fogli di stile a cascata CSS (*Cascading Style Sheets*)
 - 2.3.1. Inclusione di CSS in un documento HTML
 - 2.3.2. Commenti
 - 2.3.3. Selezionatori
 - 2.3.4. Selettori avanzati
- 2.4. Proprietà CSS (*Cascading Style Sheets*)
 - 2.4.1. Colore
 - 2.4.2. Testo
 - 2.4.3. Pseudo classi
 - 2.4.4. Transizioni
 - 2.4.5. Animazioni
 - 2.4.6. Animazione degli elementi
 - 2.4.7. Animazione avanzata
- 2.5. Box model
 - 2.5.1. Altezza e larghezza
 - 2.5.2. Margine
 - 2.5.3. Riempimento
- 2.6. Posizionamento
 - 2.6.1. Posizionamento statico
 - 2.6.2. Posizionamento relativo
 - 2.6.3. Posizionamento assoluto
 - 2.6.4. Posizionamento fisso
 - 2.6.5. *Float*
- 2.7. Progettazione adattabile
 - 2.7.1. *Viewport*
 - 2.7.2. *Media Queries*
 - 2.7.3. Unità CSS
 - 2.7.4. *Image*
 - 2.7.5. *Framework*
- 2.8. *Layout* moderno
 - 2.8.1. *Flex*
 - 2.8.2. *Grid*
 - 2.8.3. *Flex* vs. *Grid*
- 2.9. Preprocessore
 - 2.9.1. Sass
 - 2.9.2. Variabili
 - 2.9.3. Mixins
 - 2.9.4. Loop
 - 2.9.5. Funzioni
- 2.10. Sistema di Progettazione
 - 2.10.1. Bootstrap
 - 2.10.2. Griglia Bootstrap
 - 2.10.3. *Header* e *Footer* del nostro sito
 - 2.10.4. Moduli
 - 2.10.5. *Cards*
 - 2.10.6. Modalità

Modulo 3. Linguaggio JAVASCRIPT applicato al *Full Stack Developer*

- 3.1. Tipi primitivi e operatori
 - 3.1.1. Linguaggio JavaScript
 - 3.1.2. Numeri e loro operatori
 - 3.1.3. Stringhe di testo e loro operatori
 - 3.1.4. Valori booleani
 - 3.1.5. Conversioni di tipo
- 3.2. Controllori di flusso e struttura
 - 3.2.1. Espressioni e sentenze
 - 3.2.2. Variabili e costanti
 - 3.2.3. Sentenza *if*
 - 3.2.4. Sentenze *For, While*
- 3.3. Funzioni
 - 3.3.1. Funzioni
 - 3.3.2. Parametri
 - 3.3.3. Funzioni come parametri
 - 3.3.4. Ambito delle variabili
 - 3.3.5. *Scope* annidati
 - 3.3.6. *Hoisting*
 - 3.3.7. *Closure*
 - 3.3.8. Ricorsività
- 3.4. Strutture dati: oggetti
 - 3.4.1. Tipo *Object*
 - 3.4.2. Creazione di oggetti
 - 3.4.3. Accesso ai valori di un oggetto
 - 3.4.4. Aggiunta o rimozione di proprietà
 - 3.4.5. Oggetti annidati
 - 3.4.6. *Destructuring* di oggetti
 - 3.4.7. Metodi del tipo *Object*
 - 3.4.8. *Spread Operator*
 - 3.4.9. Inmutabilità
- 3.5. Strutture dati: *Array*
 - 3.5.1. Struttura dati. *Array*
 - 3.5.2. *Array*. Tipologia
 - 3.5.3. *Arrays* annidati
 - 3.5.4. Metodi di un *Array*
- 3.6. POO: *Prototype* e classi
 - 3.6.1. POO. Programmazione orientata agli oggetti
 - 3.6.2. Prototipi
 - 3.6.3. Classi
 - 3.6.4. Dati privati
 - 3.6.5. Sottoclassi
 - 3.6.6. *Call e Apply*
- 3.7. Tipi di JavaScript
 - 3.7.1. *Set*
 - 3.7.2. *WeakSet*
 - 3.7.3. *Map*
 - 3.7.4. *WeakMap*
 - 3.7.5. Espressioni regolari
- 3.8. Utilità di JavaScript
 - 3.8.1. *Date*
 - 3.8.2. *Math*
 - 3.8.3. *Symbol*
 - 3.8.4. JSON
- 3.9. JavaScript nel *Browser*
 - 3.9.1. Inclusione di JavaScript in un web
 - 3.9.2. DOM
 - 3.9.3. Eventi
 - 3.9.4. *Storage* nel navigatore
- 3.10. Programmazione asincrona
 - 3.10.1. Programmazione asincrona
 - 3.10.2. *Event Loop*
 - 3.10.3. *Callbacks*
 - 3.10.4. *Promises*
 - 3.10.5. *Async/Await*

Modulo 4. Layout web applicato al *Full Stack Developer*

- 4.1. CSS e layout
 - 4.1.1. Layout con tabelle
 - 4.1.2. Design fluido
 - 4.1.3. L'era del *Responsive*
 - 4.1.4. *Mobile First* vs. *Desktop First*
- 4.2. CSS e le regole del web design
 - 4.2.1. Selezionatori
 - 4.2.2. Pseudo classi
 - 4.2.3. Pseudo elementi
- 4.3. Layout con i CSS
 - 4.3.1. Regole del *Box Model*
 - 4.3.2. Tipografie
 - 4.3.3. Colori
 - 4.3.4. Immagini
 - 4.3.5. Sfondi
 - 4.3.6. Tabelle
 - 4.3.7. Moduli
 - 4.3.8. Mostrare e nascondere gli elementi
 - 4.3.9. Variabili CSS
- 4.4. Design *Responsive* y design fluido
 - 4.4.1. Elementi sospesi
 - 4.4.2. *Grid* CSS
 - 4.4.3. *Media Queries*
 - 4.4.4. *Flex Box*
- 4.5. La cascata CSS
 - 4.5.1. Priorità delle regole CSS
 - 4.5.2. Sovrascrittura delle regole
 - 4.5.3. Classi vs. Identificatori
- 4.6. SaSS
 - 4.6.1. Software come servizio (SaSS)
 - 4.6.2. Installazione di SaSS
 - 4.6.3. Esecuzione e compilazione di SaSS
 - 4.6.4. Struttura di una directory SaSS
- 4.7. Uso di SaSS
 - 4.7.1. Variabili in Sass
 - 4.7.2. Modulazione del nostro progetto
 - 4.7.3. Sintassi SaSS
- 4.8. Logica SaSS
 - 4.8.1. Mixins
 - 4.8.2. Maps
 - 4.8.3. Funzioni e strutture di controllo
- 4.9. Layout con Bootstrap
 - 4.9.1. Bootstrap
 - 4.9.2. *Layout* di Bootstrap
 - 4.9.3. Moduli
 - 4.9.4. *Box Model* con Bootstrap
 - 4.9.5. Colori e caratteri
 - 4.9.6. Link e pulsanti
 - 4.9.7. Mostrare e nascondere elementi con Bootstrap
 - 4.9.8. *Flex Box* con Bootstrap
 - 4.9.9. Componenti
- 4.10. *Theming* Bootstrap
 - 4.10.1. Riscrivere Bootstrap con SaSS (*Software as a Service*)
 - 4.10.2. Struttura dei file
 - 4.10.3. Creare il proprio *Framework* CSS (*Cascading Style Sheets*)

Modulo 5. Strumenti JavaScript. Libreria ReactJs

- 5.1. Strumentazione Javascript ReactJS
 - 5.1.1. Lo strumento ReactJS
 - 5.1.2. Create React App
 - 5.1.3. JavaScript *Syntax Extension*
- 5.2. Componenti di ReactJS
 - 5.2.1. Componenti
 - 5.2.2. *Props*
 - 5.2.3. Rendering
- 5.3. Eventi nella libreria ReactJS
 - 5.3.1. Gestione degli eventi
 - 5.3.2. Gestione degli eventi online
 - 5.3.3. Eventi nella libreria ReactJS
- 5.4. Configurare *Hooks* di ReactJS
 - 5.4.1. Stati di un componente
 - 5.4.2. *Hook* di stato
 - 5.4.3. *Hook* di effetto
 - 5.4.4. *Custom Hooks*
 - 5.4.5. Altri *Hooks*
- 5.5. *Component Context* in ReactJS
 - 5.5.1. *Component Context* in ReactJS
 - 5.5.2. Uso di *Context*
 - 5.5.3. Struttura del *Context*
 - 5.5.4. *React. Create Context*
 - 5.5.5. *Context. Provider*
 - 5.5.6. *Class. Context Type*
 - 5.5.7. *Context. Consumer*
 - 5.5.8. *Context.displayName*
 - 5.5.9. Applicazione pratica di *Context*
- 5.6. Routing in ReactJs
 - 5.6.1. *Router*
 - 5.6.2. *React Router*
 - 5.6.3. Installazione
 - 5.6.4. Routing di base
 - 5.6.5. Routing dinamico
 - 5.6.6. Componenti primari
 - 5.6.7. *React Router Hooks*
- 5.7. Utilizzo di elenchi e moduli con ReactJS
 - 5.7.1. Elenchi e loop
 - 5.7.2. Moduli e convalide
 - 5.7.3. *React Hook Forms*
- 5.8. Uso de stili in ReactJS
 - 5.8.1. Stiling tradizionale
 - 5.8.2. Stiling online
 - 5.8.3. Aggiunta di una libreria di sistemi di progettazione
- 5.9. Esecuzione di test in Javascript. Strumenti
 - 5.9.1. *Testing*
 - 5.9.2. *Jest JavaScript Testing Framework*
 - 5.9.3. *Visual Testing and Documentation*
- 5.10. Deployment di codice con ReactJS
 - 5.10.1. *Hosting*
 - 5.10.2. Fornitori
 - 5.10.3. Preparazione del nostro progetto
 - 5.10.4. Deployment in Heroku

Modulo 6. Framework di JavaScript. Angular

- 6.1. Il Framework angular e la sua architettura
 - 6.1.1. Angular CLI
 - 6.1.2. Architettura
 - 6.1.3. *Workspace* e struttura
 - 6.1.4. Ambiente
- 6.2. Componenti del Framework Angular
 - 6.2.1. Ciclo di vita
 - 6.2.2. Incapsulamento della vista
 - 6.2.3. Interazione tra i componenti
 - 6.2.4. Proiezione del contenuto
- 6.3. Template del Framework Angular
 - 6.3.1. Interpolazione del testo
 - 6.3.2. Dichiarazioni
 - 6.3.3. *Property Binding*
 - 6.3.4. *Class, Style e Attribute Binding*
 - 6.3.5. *Event Binding* e *Two-Way Binding*
 - 6.3.6. *Pipes*
- 6.4. Direttive del Framework Angular
 - 6.4.1. Direttive di Angular
 - 6.4.2. Direttive di attributo
 - 6.4.3. Direttive strutturali
- 6.5. Servizi e iniezione di dipendenza
 - 6.5.1. Servizi
 - 6.5.2. Immissione di dipendenze
 - 6.5.3. *Service Providers*
- 6.6. *Routing* e navigazione
 - 6.6.1. Applicazione con *Routing*
 - 6.6.2. Routing di base
 - 6.6.3. Router annidati
 - 6.6.4. Parametri
 - 6.6.5. Accesso e autorizzazione
 - 6.6.6. *Lazy Loading* di moduli
- 6.7. RxJS
 - 6.7.1. Observer
 - 6.7.2. *Observers*
 - 6.7.3. Abbonamenti
 - 6.7.4. Operatori
- 6.8. Moduli e HTTP
 - 6.8.1. Moduli reattivi
 - 6.8.2. Convalida dei campi
 - 6.8.3. Moduli dinamici
 - 6.8.4. Petizioni
 - 6.8.5. *Interceptors*
 - 6.8.6. Sicurezza
- 6.9. Animazioni
 - 6.9.1. Transizioni e *Triggers*
 - 6.9.2. Transizioni di percorsi
 - 6.9.3. Differenze tra le transizioni
- 6.10. *Testing* nel Framework Angular
 - 6.10.1. Test dei servizi
 - 6.10.2. Test dei componenti
 - 6.10.3. Test di direttive e *pipes*

Modulo 7. Programmazione in linguaggio NodeJS

- 7.1. NodeJS e la sua architettura
 - 7.1.1. NPM e gestione dei pacchetti
 - 7.1.2. Esecuzione di un programma
 - 7.1.3. Moduli
 - 7.1.4. Creazione di uno modulo
 - 7.1.5. *Loop* di eventi
- 7.2. Servidor Backend, HTTP, *Express* e *Sockets*
 - 7.2.1. Modulo HTTP
 - 7.2.2. *Express*
 - 7.2.3. *Socket.io*
- 7.3. Database e cache
 - 7.3.1. MongoDB
 - 7.3.2. Mongoose
 - 7.3.3. SQL
 - 7.3.4. *Sequelize*
 - 7.3.5. *Redis*
- 7.4. File system e *Os*
 - 7.4.1. *File System Module*
 - 7.4.2. *Os Module*
 - 7.4.3. *Cluster Module*
- 7.5. Eventi, *Buffers* e *Streams*
 - 7.5.1. Eventi
 - 7.5.2. *Buffers*
 - 7.5.3. *Streams*
- 7.6. *Testing*
 - 7.6.1. *Jest*
 - 7.6.2. *Mocha*
 - 7.6.3. TDD - *Cucumber*
- 7.7. Architettura e best practice
 - 7.7.1. DRY
 - 7.7.2. SOLID
 - 7.7.3. CRUD
 - 7.7.4. MVC
 - 7.7.5. Monoliti
 - 7.7.6. Microservizi
 - 7.7.7. Architetture esagonali
- 7.8. *Typescript*
 - 7.8.1. Tipi, interfacce e classi
 - 7.8.2. Funzioni e moduli
 - 7.8.3. Generici
 - 7.8.4. *Namespaces*
 - 7.8.5. Decoratori
- 7.9. API REST
 - 7.9.1. Get
 - 7.9.2. Post
 - 7.9.3. Put
 - 7.9.4. Delete
 - 7.9.5. *Swagger*
 - 7.9.6. Creare un'API REST con *Express*
- 7.10. Creazione e containerizzazione di un'applicazione con *NestJS*
 - 7.10.1. *Nest CLI*
 - 7.10.2. *Docker*
 - 7.10.3. Costruzione di una applicazione

Modulo 8. Database per sviluppatori *Full Stack*

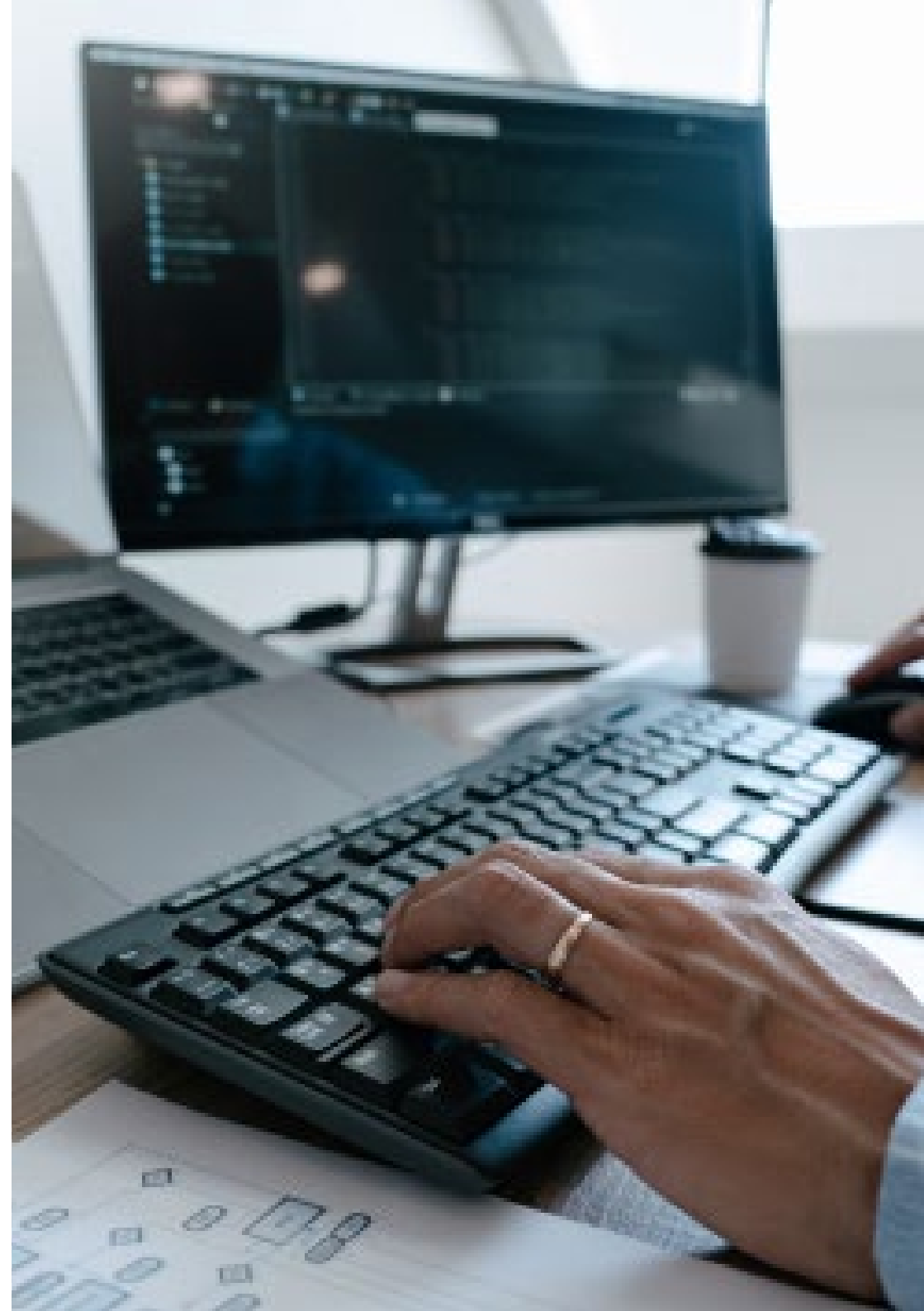
- 8.1. Database per sviluppatori *Full Stack*
 - 8.1.1. Database nello sviluppo di applicazioni
 - 8.1.2. Capacità del database
 - 8.1.3. SQL (*Structured Query Language*)
- 8.2. Scelta del database
 - 8.2.1. Applicazione o servizio da considerare
 - 8.2.2. Categorie del database
 - 8.2.3. Panorami del database
- 8.3. Sviluppo con MySQL
 - 8.3.1. Sviluppo con MySQL
 - 8.3.2. Deployment del modello relazionale con MySQL
 - 8.3.3. Connessione a MySQL
- 8.4. Sviluppo con Oracle Database
 - 8.4.1. Sviluppo con Oracle DB
 - 8.4.2. Deployment del modello
 - 8.4.3. Connessione al database Oracle
- 8.5. Sviluppo con Oracle SQL Server
 - 8.5.1. Oracle SQL Server
 - 8.5.2. Deployment del modello
 - 8.5.3. Connessione a SQL Server
- 8.6. Sviluppo con NoSQL
 - 8.6.1. Confronto con i database SQL
 - 8.6.2. Creazione del database in MongoDB
 - 8.6.3. Connessione a MongoDB
- 8.7. Sviluppo con grafi
 - 8.7.1. Sviluppo con grafi
 - 8.7.2. Creazione del database con Neo4j
 - 8.7.3. Connessione a Neo4j

- 8.8. Sviluppo con Database a valore-chiave
 - 8.8.1. Sviluppo con Database a k-v
 - 8.8.2. Creazione del database con Redis
 - 8.8.3. Connessione con Redis
- 8.9. Database con altri tipi di dati
 - 8.9.1. *Elastic Search*
 - 8.9.2. *Inmemory Database*
 - 8.9.3. Sviluppo con dati spaziali
- 8.10. Database. Aspetti avanzati
 - 8.10.1. I database negli sviluppi cloud native
 - 8.10.2. I database nell'architettura a microservizi
 - 8.10.3. CI/CD e i database

Modulo 9. UX CX. Esperienza del cliente

- 9.1. *Customer Experience*
 - 9.1.1. *Customer Experience (CX)*
 - 9.1.2. Nuove esigenze dei consumatori
 - 9.1.3. *Feedback* nel *Customer Experience*
- 9.2. Tecnologia innovatrice
 - 9.2.1. Macchine che pensano
 - 9.2.2. Nuove modalità di condivisione delle informazioni
 - 9.2.3. Misurare ciò che non si può misurare
- 9.3. Canali di interazione con l'utente
 - 9.3.1. Analisi dei clienti
 - 9.3.2. Personalizzazione
 - 9.3.3. Multipli canali di interazione con l'utente

- 9.4. Analitica degli utenti
 - 9.4.1. Struttura del sito web
 - 9.4.2. Analitica degli utenti
 - 9.4.3. Analitica degli utenti avanzata
- 9.5. Nielsen e il suo impatto sulla CX
 - 9.5.1. Nielsen e il suo impatto sulla CX
 - 9.5.2. Tecniche di *User Testing*
- 9.6. Strumenti di *Customer Experience*
 - 9.6.1. Strumenti avanzati
 - 9.6.2. Mobilità
 - 9.6.3. Accessibilità
- 9.7. Nuove metodologie
 - 9.7.1. Le sfide dell'utente
 - 9.7.2. Processo UX
 - 9.7.3. Ricerca sugli utenti
- 9.8. Comunicare un progetto
 - 9.8.1. *Wireframing*
 - 9.8.2. Strumenti di comunicazione per la progettazione
 - 9.8.3. Strumenti avanzati di comunicazione per la progettazione
- 9.9. Progettazione UI
 - 9.9.1. Progettazione UI
 - 9.9.2. Interfacce web e mobile
 - 9.9.3. Componenti web e mobile
- 9.10. Preparazione di un CX
 - 9.10.1. Preparazione di un CX
 - 9.10.2. Progettazione di nuove esperienze
 - 9.10.3. Interfaccia



Modulo 10. Integrazione continua e deployment delle applicazioni

- 10.1. Integrazione e deployment continuo: CI/CD
 - 10.1.1. Uso di integrazione e deployment continuo (CI/ CD)
 - 10.1.2. Differenze tra integrazione e deployment continuo (CI/ CD)
 - 10.1.3. Integrazione e deployment continuo. Benefici del CI/CD
- 10.2. Nuovi paradigmi di sviluppo
 - 10.2.1. Da Waterfall a DevOps
 - 10.2.2. Guida di stile: i 12 fattori
 - 10.2.3. Cloud Native, microservizi e *Serverless*
- 10.3. DevOps, oltre il CI/CD
 - 10.3.1. DevOps
 - 10.3.2. DevOps. *Continuous Everything*
 - 10.3.3. DevOps vs SRE
- 10.4. Tecnologia dei container I - Docker
 - 10.4.1. Container. Contributi
 - 10.4.2. Docker. Architettura
 - 10.4.3. Processo di deployment con Docker
- 10.5. Tecnologia dei container II - Kubernetes
 - 10.5.1. Orchestrazione
 - 10.5.2. Kubernetes
 - 10.5.3. L'ecosistema di Kubernetes
- 10.6. Configurazione dell'infrastruttura con GitOps
 - 10.6.1. Infrastruttura immutabile
 - 10.6.2. GitOps
 - 10.6.3. Strumenti GitOps
- 10.7. Pipeline e automazione. Casi d'Uso CI/CD
 - 10.7.1. Integrazione continua
 - 10.7.2. Deployment e continuous delivery
 - 10.7.3. Convalide automatiche
 - 10.7.4. Best practice di CI/CD
- 10.8. CI/CD con Jenkins. Riferimenti
 - 10.8.1. CI/CD con Jenkins
 - 10.8.2. Pipelines Jenkins
 - 10.8.3. Best practice con Jenkins
- 10.9. Ecosistema CI/ CD
 - 10.9.1. Organizzazione dell'ecosistema
 - 10.9.2. Strumenti avanzati
 - 10.9.3. Dagger. Il futuro
- 10.10. Fasi finali del ciclo del software orientato al CI/CD
 - 10.10.1. Applicazione dell'IA al processo IC/DC
 - 10.10.2. DevSecOps
 - 10.10.3. *Chaos Engineering*



Avrai a disposizione le più moderne risorse didattiche, con accesso gratuito all'aula virtuale 24 ore su 24"

06

Metodologia

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

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



“

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

Caso di Studio per contestualizzare tutti i contenuti

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

“

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



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



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

Un metodo di apprendimento innovativo e differente

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

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

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

Cosa dovrebbe fare un professionista per affrontare una determinata situazione?

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

Metodologia Relearning

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

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

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

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

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



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

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

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

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

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



Materiali di studio

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

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



Master class

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

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



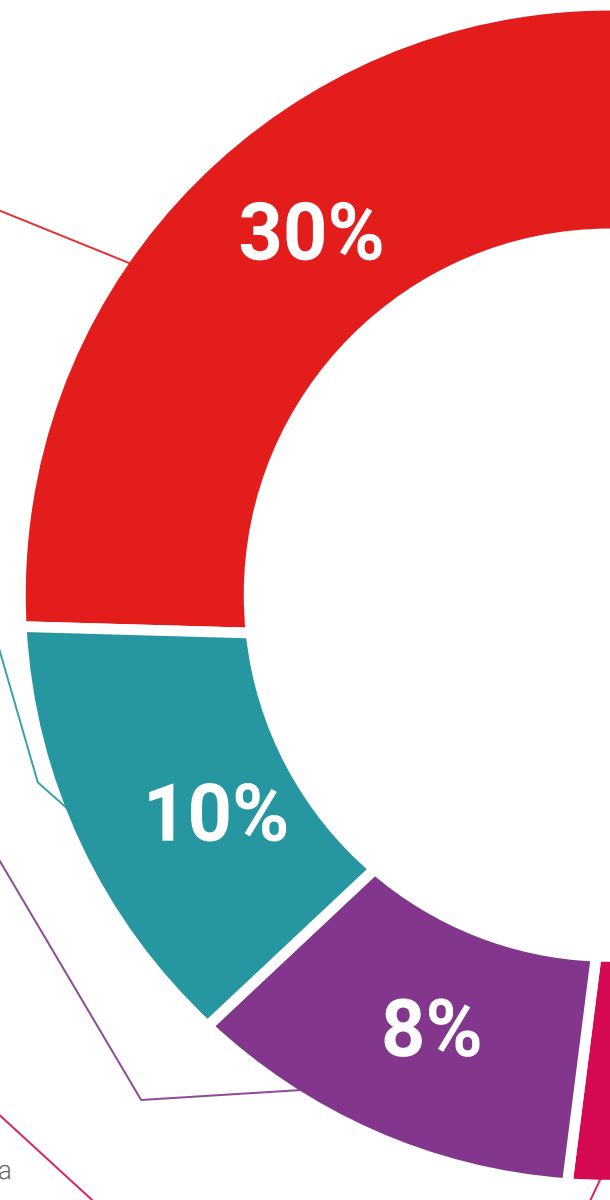
Pratiche di competenze e competenze

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



Letture complementari

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





Casi di Studio

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



Riepiloghi interattivi

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

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



Testing & Retesting

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



07 Titolo

Il Master Privato in Programmazione Front End Full Stack Developer ti garantisce, oltre alla preparazione più rigorosa e aggiornata, l'accesso a una qualifica di Master Privato rilasciata da TECH Università Tecnologica.



“

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

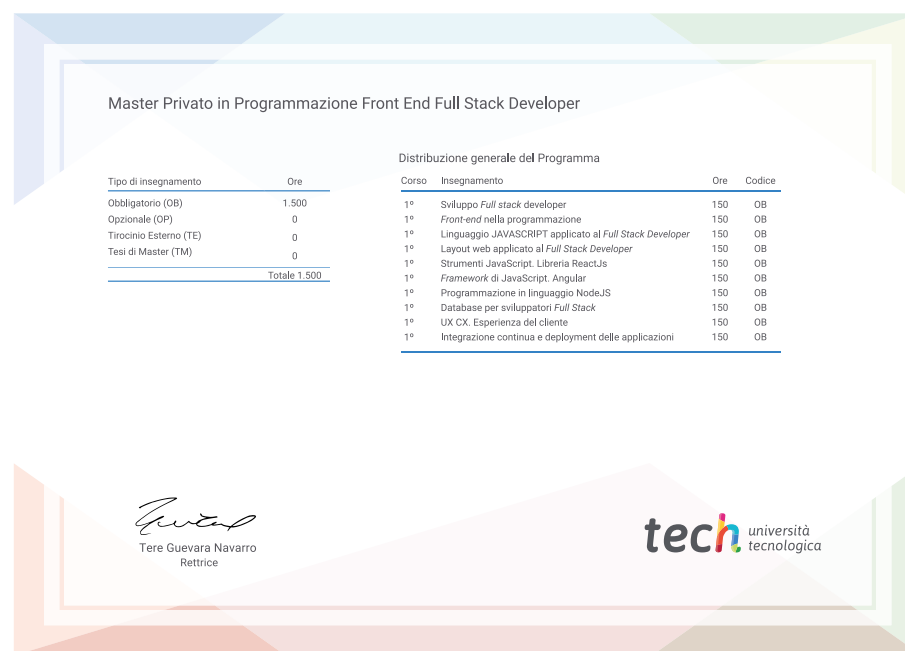
Questo **Master Privato in Programmazione Front End Full Stack Developer** possiede il programma più completo e aggiornato del mercato.

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

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

Titolo: **Master Privato in Programmazione Front End Full Stack Developer**

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



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

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

tech università
tecnologica

Master Privato
Programmazione Front
End Full Stack Developer

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

Master Privato

Programmazione Front
End Full Stack Developer