

Master Semipresenziale

Programmazione Front End Full Stack Developer



Master Semipresenziale

Programmazione Front End Full Stack Developer

Modalità: Semipresenziale (Online + Tirocinio)

Durata: 12 mesi

Titolo: TECH Global University

Crediti: 60 + 4 ECTS

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

Indice

01

Presentazione

pag. 4

02

Perché iscriversi a questo
Master Semipresenziale?

pag. 8

03

Obiettivi

pag. 12

04

Competenze

pag. 18

05

Direzione del corso

pag. 22

06

Struttura e contenuti

pag. 26

07

Tirocinio

pag. 38

08

Dove posso svolgere il
Tirocinio?

pag. 44

09

Metodologia

pag. 48

10

Titolo

pag. 56

01

Presentazione

Le aziende cercano sviluppatori che non solo abbiano la capacità di creare interfacce utente accattivanti e funzionali, ma che siano anche in grado di sviluppare server e database. Con il boom di tecnologie come React, Node.js e la crescente importanza dell'esperienza utente (UX), è possibile integrare più livelli di sviluppo web sono altamente apprezzati. Infatti, con l'evoluzione costante delle metodologie agili e dell'automazione dei processi, i *Full Stack Developers* devono essere sempre aggiornati. In questo contesto, TECH ha sviluppato questo programma completo, che si divide in due aree, una prima completamente online, basata sulla metodologia innovativa *Relearning*, e un tirocino operativo di 3 settimane presso una rinomata azienda.



“

*Grazie a questo Master Semipresenziale,
padroneggerai le tecniche di programmazione,
acquisendo anche competenze in gestione di
progetti, esperienza utente e sviluppo continuo”*

La domanda di professionisti nella Programmazione Front End Full Stack Developer continua a crescere, spinta dall'accelerazione della trasformazione digitale in più settori. Con la proliferazione di applicazioni web e mobile, le aziende sono alla ricerca di sviluppatori in grado di gestire sia il *front-end* che il *back-end*.

Nasce così questo Master Semipresenziale, che sarà diviso in due sezioni distinte. In primo luogo, in una parte teorica che fornirà una panoramica delle tecnologie e le competenze necessarie per lo sviluppo completo di applicazioni web, tra cui un focus sul *front-end* in Programmazione, dove gli informatici si concentreranno sulle tecniche e le migliori pratiche per creare interfacce utente efficienti e accattivanti.

Il piano di studi sarà inoltre approfondito nel linguaggio JavaScript applicato a *Full Stack Developer*, cruciale per qualsiasi sviluppatore moderno, nonché nel layout web applicato, dove i professionisti struttureranno e stilizzeranno le pagine web in modo professionale. Inoltre, gli strumenti JavaScript e le librerie specifiche come ReactJs saranno indispensabili per lo sviluppo di applicazioni dinamiche e scalabili.

Inoltre, si approfondirà la programmazione in NodeJS e nei database per Sviluppatori *Full Stack*, compresa la logica del server e la gestione dei database. Verrà anche affrontata l'importanza dell'esperienza utente, UX CX, e si concluderà con l'integrazione continua e la distribuzione delle applicazioni. Infine, la seconda parte della formazione consisterà in un'intensa esperienza pratica di 3 settimane presso una delle migliori aziende informatiche, accuratamente selezionate per portare la carriera dello studente al livello successivo.

In questo modo, TECH ha progettato un programma completamente online e totalmente adattabile che richiede solo un dispositivo elettronico con connessione a Internet per accedere ai materiali didattici. Inoltre, l'area teorica si baserà sulla rivoluzionaria metodologia *Relearning*, pioniera in questa università e consistente nella ripetizione di concetti chiave per un'assimilazione ottimale e organica dei contenuti.

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

- ♦ Sviluppo di oltre 100 casi pratici presentati da professionisti del settore dell'Informatica, esperti di Programmazione Front End Full Stack Developer, oltre professori universitari con una vasta esperienza in materia
- ♦ Contenuti grafici, schematici ed eminentemente pratici che forniscono informazione tecnica riguardo alle discipline essenziali per l'esercizio della professione
- ♦ Lezioni teoriche, domande all'esperto, forum di discussione su argomenti controversi e lavori di riflessione individuale
- ♦ Contenuti disponibili da qualsiasi dispositivo fisso o mobile dotato di connessione a internet
- ♦ Possibilità di svolgere un tirocinio presso una delle migliori aziende del settore



Approfondirai la creazione di interfacce utente, con tecniche avanzate di HTML, CSS e JavaScript per progettare esperienze utente intuitive e reattive. Cosa aspetti ad iscriverti?"

“

Ti concentrerai sul linguaggio JavaScript applicato a Full Stack Developer, padroneggiando JavaScript, la spina dorsale della maggior parte delle moderne applicazioni web. Con tutte le garanzie di qualità!”

In questa proposta di Master, di natura professionalizzante e in modalità Semipresenziale, il programma è rivolto all'aggiornamento degli informatici che svolgono le loro funzioni in aziende specializzate nello sviluppo di software,

e che richiedono un alto livello di qualifica. I contenuti sono basati sulle ultime evidenze scientifiche, orientati in modo didattico per integrare le conoscenze teoriche nella pratica informatica, e gli elementi teorico-pratici faciliteranno l'aggiornamento delle conoscenze e permetteranno di prendere decisioni in situazioni complesse.

Grazie ai contenuti multimediali sviluppati in base all'ultima tecnologia educativa, si consentirà al professionista sanitario di ottenere un apprendimento situato e contestuale, ovvero un ambiente simulato che fornirà un insegnamento immersivo programmato per affrontare situazioni reali. 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.

Svolgi un tirocinio di 3 settimane presso un centro prestigioso e acquisisci tutte le conoscenze di cui hai bisogno per crescere personalmente e lavorativamente.

Analizzerai i principi del design incentrato sull'utente, migliorando l'esperienza del cliente, attraverso i migliori materiali didattici del mercato accademico, all'avanguardia tecnologica ed educativa.



“

Avrai l'opportunità di imparare attraverso una metodologia mista, che include lezioni online e un tirocinio pratico innovativo, facilitando un'esperienza di apprendimento più dinamica e pratica"

1. Aggiornarsi sulla base delle più recenti tecnologie disponibili

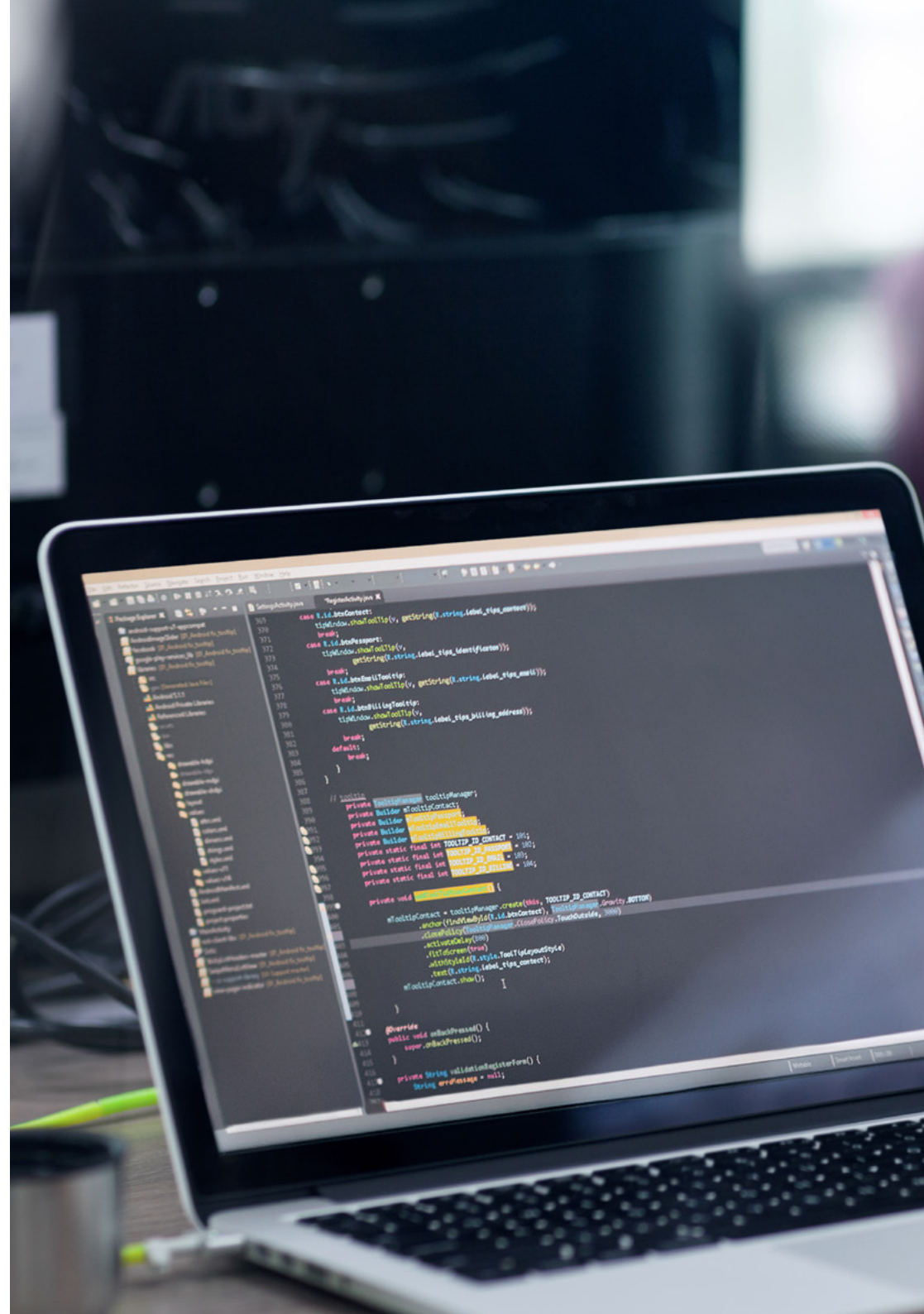
La più recente tecnologia di Programmazione Front End Full Stack Developer è caratterizzata dall'integrazione di strumenti e *framework* avanzati che facilitano lo sviluppo di applicazioni web più veloci, sicure e scalabili. Nel *front-end*, *framework* come React e Angular continuano a evolversi, offrendo miglioramenti in termini di prestazioni e funzionalità. Nel *back-end*, invece, Node.js rimane uno strumento fondamentale. Queste innovazioni non solo migliorano la produttività degli sviluppatori, ma elevano anche lo standard delle applicazioni web in termini di prestazioni e usabilità.

2. Approfondire nuove competenze dall'esperienza dei migliori specialisti

Il team di professionisti che accompagnerà lo specialista per tutto il periodo di pratica rappresenta un'assicurazione di prima classe e una garanzia di aggiornamento senza precedenti. Con un tutor appositamente nominato, l'informatico sarà in grado di vedere situazioni reali in un ambiente all'avanguardia, che gli permetterà di incorporare nella sua pratica quotidiana le procedure e gli strumenti più efficaci in Programmazione Front End Full Stack Developer.

3. Accedere ad ambienti professionali di prim'ordine

TECH seleziona con cura tutti i centri disponibili per lo svolgimento del Tirocinio. Allo specialista viene garantito l'accesso in un ambiente clinico di prestigio e specializzato nella Programmazione Front End Full Stack Developer. In questo modo, lo studente avrà l'opportunità di sperimentare il lavoro quotidiano di un settore esigente, rigoroso ed esaustivo, applicando sempre le tesi e i postulati scientifici più recenti nella propria metodologia di lavoro.



4. Combinare la migliore teoria con la pratica più avanzata

Il mercato accademico è pieno di programmi educativi poco adattabili alle attività quotidiane dello specialista e che richiedono lunghe ore di carico didattico, spesso poco compatibili con la vita personale e professionale. TECH offre un nuovo modello di apprendimento, in modalità 100% pratica, che permette di conoscere le procedure più avanzate in materia di Programmazione Front End Full Stack Developer e, soprattutto, di metterle in pratica in sole 3 settimane.

5. Accedere a nuove opportunità

Gli sviluppatori *Full Stack*, con la loro capacità di gestire sia il *front-end* che il *back-end* di un'applicazione, sono altamente apprezzati per la loro versatilità

e l'efficienza nella creazione di soluzioni complete. Questa padronanza completa consente di lavorare su una varietà di progetti, dalle startup innovative alle grandi aziende tecnologiche e adattarsi a diversi ruoli, sia come sviluppatore, architetto del software o leader tecnico. Con la crescente attenzione alla digitalizzazione e alla trasformazione digitale, gli sviluppatori *Full Stack* sono in una posizione privilegiata per sfruttare queste opportunità e far progredire le loro carriere.

“Avrai l'opportunità di svolgere un tirocinio all'interno di un centro a tua scelta”

03

Obiettivi

Questo programma universitario cercherà di fornire agli informatici una conoscenza approfondita e pratica delle tecnologie più avanzate nella programmazione web, come HTML, CSS, JavaScript, React, Angular, Node.js e database relazionali e non relazionali. Inoltre, saranno sviluppate competenze in metodologie agili, gestione dei progetti e strumenti di integrazione e distribuzione continua, garantendo che gli studenti possano adattarsi alle richieste dinamiche del settore tecnologico. Un altro obiettivo chiave sarà quello di infondere un approccio incentrato sull'utente, attraverso i principi UX e CX per creare esperienze digitali eccezionali.



“

Al termine di questo Master Semipresenziale sarai preparato ad affrontare con successo le sfide del mercato del lavoro, evidenziando la tua versatilità, capacità di innovazione e competenze tecniche avanzate"



Obiettivo generale

- ♦ L'obiettivo generale che ha il Master Semipresenziale in Programmazione Front End Full Stack Developer sarà quello di fornire una conoscenza specializzata in aspetti chiave della programmazione, focalizzata sui fondamenti essenziali del web, compresa la corretta sintassi dei linguaggi HTML e CSS. Inoltre, i professionisti saranno addestrati a sviluppare qualsiasi tipo di applicazione utilizzando JavaScript, compreso l'uso della libreria Bootstrap e l'implementazione di progetti di layout con SaSS. Inoltre, saranno sviluppate competenze avanzate nell'uso di *framework* moderni come React e Angular, esplorando il potenziale del linguaggio NodeJS



Gestirai database relazionali e non relazionali, essenziali per l'archiviazione e la manipolazione dei dati, da parte della migliore università digitale del mondo, secondo Forbes: TECH"





Obiettivi specifici

Modulo 1. Sviluppatore *Full Stack Developer*

- ♦ Sviluppare conoscenze avanzate di programmazione
- ♦ Incoraggiare l'uso di sistemi di controllo delle versioni e di piattaforme di hosting del codice
- ♦ Promuovere l'uso di metodologie agili
- ♦ Approfondire i concetti chiave e il funzionamento di Internet
- ♦ Aumentare la competenza della linea di comando

Modulo 2. *Front-end* in Programmazione

- ♦ Identificare e comprendere la sintassi corretta di HTML e CSS
- ♦ Esplorare i diversi elementi dell'HTML
- ♦ Determinare l'approccio di progettazione adattativa
- ♦ Utilizzare la presentazione delle 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 a *Full Stack Developer*

- ♦ Stabilire i tipi di base e complessi offerti da JavaScript
- ♦ Analizzare i diversi modi di programmare con il linguaggio e fare l'uso corretto in ogni situazione
- ♦ Aggiornare le conoscenze sulle ultime versioni
- ♦ Scoprire la programmazione funzionale
- ♦ Esaminare la programmazione asincrona e le sue caratteristiche

Modulo 4. Layout Web applicato a *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*
- ♦ Approfondire i fondamenti dei principi di sviluppo a cascata dei CSS
- ♦ Identificare la tecnologia *Bootstrap* in qualsiasi progetto web
- ♦ Analizzare i principi di *Bootstrap*
- ♦ Sviluppare un layout web utilizzando *Bootstrap*
- ♦ Determinare i principi di sviluppo in un progetto SaSS

Modulo 5. Strumenti di JavaScript Libreria ReactJs

- ♦ Determinare le funzionalità di React
- ♦ Configurare un progetto utilizzando Create React App
- ♦ Analizzare il ciclo di vita dei componenti in React
- ♦ Generare competenze sulle moderne funzionalità di React come *Hooks* e *Context*
- ♦ Impostare gli stati globali utilizzando il contesto
- ♦ Creare e renderizzare elenchi e creare moduli con React
- ♦ Implementare la convalida dei campi nei moduli
- ♦ Stilizzare componenti ed elementi
- ♦ Eseguire debug, test e deployment di applicazioni React

Modulo 6. *Framework* di JavaScript: Angular

- ♦ Sviluppare conoscenze specialistiche sull'architettura del *Framework*
- ♦ Approfondire la metodologia 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 Node.js

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 di omnicanalità e personalizzazione basate su microinterazioni
- ♦ Studiare l'evoluzione della web analitica fino all'analitica comportamentale
- ♦ Determinare come l'intelligenza artificiale ha portato la CX al livello successivo
- ♦ 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
- ♦ Discutere strumenti concreti di prototipazione e *wireframing*, nonché *frameworks* di sviluppo *front-end*

Modulo 10. Integrazione continua e implementazione delle applicazioni

- ♦ Realizzare i vantaggi dell'adozione di un modello di implementazione automatica delle applicazioni
- ♦ Stabilire le differenze tra integrazione continua, consegna continua e implementazione continua
- ♦ Determinare le caratteristiche principali di DevOps
- ♦ Valutare alcuni degli strumenti principali per l'implementazione di pipeline CI/CD
- ♦ Sviluppare i fattori essenziali per lo sviluppo di applicazioni pronte a supportare i processi CI/CD
- ♦ Esaminare le tecnologie dei container come pilastro fondamentale della pratica CI/CD

04

Competenze

Le competenze chiave di questa qualifica accademica includeranno la capacità di progettare e sviluppare applicazioni web complete, utilizzando tecnologie *front-end* e *back-end* avanzati, come HTML, CSS, JavaScript, React, Angular e Node.js. Gli informatici acquisiranno anche competenze nell'implementazione di pratiche UX/UI per migliorare l'esperienza utente e nell'integrazione di database per una gestione efficiente dei dati nelle applicazioni. Inoltre, sarà enfatizzata l'uso di metodologie agili e strumenti CI/CD per garantire la fornitura continua e il miglioramento costante dei prodotti digitali.

```
...ings.js
...js
command.js
editor.js
FileManager.js
main.js
README.rst
sequences.js
```

```
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
```

```
self
input.on
//escap
if (e.
// th
self.
return
}
if (e.
e.sto
e.pro
self.
self.
return
}
//up/d
if (e.
e.pro
e.sto
if (
//
if
```

```
searchHistory;  
= this;  
("keydown", function(e) {  
  if (e.keyCode == 27) {  
    // this is a new line  
    deactivate(true);  
    return; // this is a modified line  
  }  
})  
  
if (e.keyCode == 13) {  
  stopImmediatePropagation();  
  eventDefault();  
  search();  
  deactivate();  
  return;  
}
```

```
if (e.keyCode == 13) {  
  stopImmediatePropagation();  
  eventDefault();  
  search();  
  deactivate();  
  return;  
}
```

```
if (e.keyCode == 38 || e.keyCode == 40) {  
  eventDefault();  
  stopImmediatePropagation();  
  if (e.keyCode == 38) {  
    showPrevious();  
  } else {  
    showNext();  
  }  
}
```

“

Approfondirai le pratiche DevOps, assicurando l'implementazione e la manutenzione delle applicazioni in modo efficiente e sicuro, grazie a una vasta libreria di risorse multimediali"



Competenze generali

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





Competenze specifiche

- ♦ Analizzare le diverse strutture data
- ♦ Esaminare le tecniche di progettazione e interpretazione degli algoritmi
- ♦ Preparazione dell'ambiente di sviluppo
- ♦ Clonare il sito web
- ♦ Creare 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 che sono essenziali per supportare il processo complessivo

05

Direzione del corso

Il personale docente del Master Semipresenziale in Programmazione Front End Full Stack Developer è composto da professionisti esperti e con una vasta esperienza nel campo dello sviluppo web. Infatti, sono impegnati a fornire un insegnamento di alta qualità e aggiornato, basato sulla loro profonda conoscenza di tecnologie come HTML, CSS, JavaScript, React, Angular, Node.js, e vari database. Oltre alla loro esperienza tecnica, sono abili nell'applicazione di metodologie agili e nella gestione dei progetti, garantendo che gli studenti acquisiscano sia competenze pratiche che una solida comprensione dei principi fondamentali dello sviluppo *Full Stack*.



“

L'impegno del personale docente non è solo quello di trasmettere conoscenze teoriche e pratiche, ma anche di guidarti verso l'eccellenza professionale, preparandoti a eccellere nell'ambiente di lavoro"

Direzione



Dott. Olalla Bonal, Martín

- ♦ Responsabile Senior della Pratica *Blockchain* presso EY
- ♦ Specialista Tecnico *Blockchain* Client presso IBM
- ♦ Direttore dell'Architettura di Blocknitive
- ♦ Coordinatore del Team per i Database Distribuiti Non-Relazionali per wedoIT (filiale presso IBM)
- ♦ Architetto di Infrastrutture presso Bankia
- ♦ Responsabile del Dipartimento di Layout di T-Systems
- ♦ Coordinatore del Dipartimento per Bing Data España SL

Personale docente

Dott. Calzada Martínez, Jesús

- ♦ Ingegnere software senior presso Devo
- ♦ Sviluppatore *Full Stack* presso Blocknitive
- ♦ Responsabile di *Frontend* presso Infinia
- ♦ Sviluppatore *Full Stack* presso Resem
- ♦ Sviluppatore Java presso Hitec
- ♦ Laurea in Ingegneria Informatica presso l'Università Complutense di Madrid

Dott. Frias Favero, Pedro Luis

- ♦ Architetto capo Blockchain presso EY
- ♦ Cofondatore e Direttore Tecnico di Swear IT Technologies
- ♦ Direttore del supporto IT per Key Business One in Messico, Colombia e Spagna
- ♦ Laurea in Ingegneria Industriale presso l'Università di Yacambú
- ♦ Esperto di Blockchain e Applicazioni Decentralizzate presso l'Università di Alcalá

Dott. Reyes Oliva, Luis

- ♦ Architetto di *Software e Cloud*
- ♦ Promotore di sviluppo e Architetto di *Cloud* presso IBM
- ♦ Responsabile tecnico dei clienti per i conti integrati di BBVA presso IBM
- ♦ Architetto di *Cloud e DevOps* presso IBM
- ♦ Architetto *software* per i clienti presso Telefónica
- ♦ Architetto di soluzioni tecniche presso Rational
- ♦ *Software Engineering Manager* presso Borland
- ♦ *Project Manager* presso Altana Consulting
- ♦ Laurea in Ingegneria informatica presso l'Università Pontificia di Salamanca

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
- ♦ Laureato in Ingegneria Tecnica delle Telecomunicazioni, specializzato in Sistemi Elettronici presso l'Università Politecnica di Valencia
- ♦ Laureato in Ingegneria Elettronica delle Comunicazioni presso l'Università Politecnica di Madrid

Dott. Gómez Rodríguez, Antonio

- ♦ Ingegnere Principale di Soluzioni Cloud per Oracle
- ♦ Co-organizzatore del Malaga Developer Meetup
- ♦ Consulente Specializzato presso Sopra Group e Everis
- ♦ Leader dei team presso System Dynamics
- ♦ Sviluppatore software presso SGO Software
- ♦ Master in E-Business presso la Business School La Salle
- ♦ Studi post-laurea sulle Tecnologie e i Sistemi Informatici svolti presso l'Istituto Catalano di Tecnologia
- ♦ Laurea in Ingegneria delle Telecomunicazioni presso l'Università Politecnica della Catalogna

Dott. Guerrero Díaz-Pintado, Arturo

- ♦ Direttore di Customer Experience per IBM
- ♦ Ingegnere tecnico di prevendita attraverso il *Watson Customer Engagement* portfolio
- ♦ Ingegnere di rete R&D presso Telefónica
- ♦ Laurea in Ingegneria delle Telecomunicazioni presso l'Università di Alcalá e la *Danish Technical University*

06

Struttura e contenuti

I contenuti del programma copriranno tutto, dai fondamenti essenziali come HTML, CSS e JavaScript alle tecnologie avanzate come React, Angular e Node.js per lo sviluppo *front-end* e *back-end* di applicazioni web. Approfondirà il layout web, la progettazione di interfacce e l'implementazione di pratiche UX/UI per ottimizzare l'esperienza utente. Inoltre, saranno sviluppate le tecniche di integrazione continua e di implementazione delle applicazioni (CI/CD), nonché la gestione efficiente dei database e l'applicazione di metodologie agili per la gestione dei progetti.

```
...etr.getString  
...if (settings[0]  
...if (name.compare  
...name += "  
...}  
...name += Date  
...if (set
```

“

Il Master Semipresenziale in Programmazione Front End Full Stack Developer coprirà una vasta gamma di contenuti, progettati per fornire una formazione completa nello sviluppo web moderno"

Modulo 1. Sviluppo di *Full Stack Developer*

- 1.1. Sviluppo di *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 di *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. Struttura dei dati nello sviluppo
 - 1.3.1. Tipi di struttura lineari
 - 1.3.2. Tipi di struttura funzionale
 - 1.3.3. Tipi di strutture ad albero
- 1.4. Progettazione e interpretazione di algoritmi
 - 1.4.1. Parallelismo nello sviluppo: Dividi e vincerai
 - 1.4.2. Algoritmi voraci
 - 1.4.3. Programmazione dinamica
- 1.5. Ambiente e strumenti per lo sviluppo orientato 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 Linea di Comando
 - 1.6.7. *Secure Shell*: SSH
 - 1.6.8. Comandi avanzati

- 1.7. Git: Deposito di software
 - 1.7.1. Git: Deposito di software
 - 1.7.2. Utilizzo di Git
 - 1.7.3. Deposito 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. Depositi
- 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* in 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. Primo *Site*
- 2.2. Linguaggio HTML: *Layouts*
 - 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. Modello box

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. *Floats*

2.7. Design adattabile

2.7.1. *Viewport*

2.7.2. *Media queries*

2.7.3. Unità CSS

2.7.4. *Images*

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. Sistemi di disegno

2.10.1. Bootstrap

2.10.2. Griglia di Bootstrap

2.10.3. *Header* e *footer* del nostro sito

2.10.4. Formulari

2.10.5. *Cards*

2.10.6. Modalità

Modulo 3. Linguaggio JavaScript applicato a *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 sui suoi operatori

3.1.4. Valori booleani

3.1.5. Conversione tra tipi

3.2. Controllori di flusso e struttura

3.2.1. Espressioni e frasi

3.2.2. Variabili e costanti

3.2.3. Istruzione *if*

3.2.4. Istruzione *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. *Scopes* nidificati

3.3.6. *Hoisting*

3.3.7. *Closures*

3.3.8. Risorse

- 3.4. Struttura dei dati: Obiettivi
 - 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 nidificati
 - 3.4.6. *Destructuring* di oggetti
 - 3.4.7. Metodi di tipo *Object*
 - 3.4.8. *Spread operator*
 - 3.4.9. Immutabilità
- 3.5. Struttura dei dati: *Array*
 - 3.5.1. Struttura dei dati: *Array*
 - 3.5.2. *Array*: Tipologia
 - 3.5.3. *Array* nidificati
 - 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 sito web
 - 3.9.2. DOM
 - 3.9.3. Eventi
 - 3.9.4. *Storage* nel browser
- 3.10. Programmazione Asincrona
 - 3.10.1. La 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 a *Full Stack Developer*

- 4.1. CSS e layout
 - 4.1.1. Layout con tabelle
 - 4.1.2. Layout 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 CSS
 - 4.3.1. Regole del *Box Model*
 - 4.3.2. Font
 - 4.3.3. Colori
 - 4.3.4. Immagini
 - 4.3.5. Fondi
 - 4.3.6. Tabelle
 - 4.3.7. Formolari
 - 4.3.8. Mostrare e nascondere gli elementi
 - 4.3.9. Variabili CSS

- 4.4. Design *Responsive* e fluido
 - 4.4.1. Elementi fluttuanti
 - 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 e 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 un repertorio SaSS
- 4.7. Uso di SaSS
 - 4.7.1. Variabili in Sass
 - 4.7.2. Modularizzazione del progetto
 - 4.7.3. Sintassi di SaSS
- 4.8. Logica di 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. Formulare
 - 4.9.4. *Box Model* con Bootstrap
 - 4.9.5. Colori e caratteri
 - 4.9.6. Collegamenti 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. Creazione del proprio *framework* CSS (*Cascading Style Sheets*)

Modulo 5. Strumenti di JavaScript: Libreria ReactJS

- 5.1. Strumenti di 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. Configurazione degli *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 *hook*
- 5.5. Componente *Context* in ReactJS
 - 5.5.1. Componente *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 dell'uso 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 principali
 - 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 validazioni
 - 5.7.3. *React hook forms*
- 5.8. Utilizzo degli stili in ReactJS
 - 5.8.1. Stile tradizionale
 - 5.8.2. Stile 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. Distribuzione del codice con ReactJS
 - 5.10.1. *Hosting*
 - 5.10.2. Fornitori
 - 5.10.3. Preparazione del progetto
 - 5.10.4. Distribuzione 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 circostante
- 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. Modelli 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 di struttura
- 6.5. Servizi e iniezione di dipendenza
 - 6.5.1. Servizi
 - 6.5.2. Iniezione 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. Percorsi nidificati
 - 6.6.4. Parametri
 - 6.6.5. Accesso e autorizzazione
 - 6.6.6. *Lazy loading* dei moduli

- 6.7. RxJS
 - 6.7.1. Osservabili
 - 6.7.2. *Observers*
 - 6.7.3. Iscrizioni
 - 6.7.4. Operatori
- 6.8. Formulare e HTTP
 - 6.8.1. Formulare reattivi
 - 6.8.2. Validazione dei campi
 - 6.8.3. Formulare 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 percorso
 - 6.9.3. Differenze tra le Transizioni
- 6.10. *Testing* nel *Framework* Angular
 - 6.10.1. Test di servizi
 - 6.10.2. Test dei componenti
 - 6.10.3. Test delle 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. Server 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 Buone Pratiche
 - 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. Costruzione di una API REST con Express
- 7.10. Creazione e contenimento 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 all'interno dello sviluppo di applicazioni
 - 8.1.2. Capacità dei Database
 - 8.1.3. *SQL(Structured Query Language)*
- 8.2. Scelta di Database
 - 8.2.1. Applicazione o servizio da considerare
 - 8.2.2. Categorie di database
 - 8.2.3. Panorami di database
- 8.3. Sviluppo con MySQL
 - 8.3.1. Sviluppo con MySQL
 - 8.3.2. Distribuzione 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. Distribuzione del modello
 - 8.4.3. Connessione al Database Oracle
- 8.5. Sviluppo con Oracle SQL Server
 - 8.5.1. Oracle Server SQL
 - 8.5.2. Distribuzione del modello
 - 8.5.3. Connessione a SQL Server

- 8.6. Sviluppo NoSQL
 - 8.6.1. Confronto con database SQL
 - 8.6.2. Creazione del database in MongoDB
 - 8.6.3. Connessione a MongoDB
- 8.7. Sviluppo con Grafici
 - 8.7.1. Sviluppo con Grafici
 - 8.7.2. Creazione di database con Neo4j
 - 8.7.3. Connessione a Neo4j
- 8.8. Sviluppo con database Chiave-Valore
 - 8.8.1. Sviluppo con database K-v
 - 8.8.2. Creazione di database con Redis
 - 8.8.3. Connessione a 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. Database nell'architettura a microservizi
 - 8.10.3. CI/CD e 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 nella Customer Experience*
- 9.2. Tecnologia e Innovazione
 - 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. Molteplici canali di interazione con l'utente

- 9.4. Analisi dell'utente
 - 9.4.1. Struttura del sito web
 - 9.4.2. Analisi dell'utente
 - 9.4.3. Analisi dell'utente 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. La sfida dell'utente
 - 9.7.2. Processo di UX
 - 9.7.3. Ricerca sull'utente
- 9.8. Comunicazione di un progetto
 - 9.8.1. *Wireframing*
 - 9.8.2. Strumenti di comunicazione per la progettazione
 - 9.8.3. Strumenti avanzati di comunicazione del design
- 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. Elaborazione di una CX
 - 9.10.1. Elaborazione di una CX
 - 9.10.2. Progettare nuove esperienze
 - 9.10.3. Interfacce

Modulo 10. Integrazione Continua e Implementazione delle Applicazioni

- 10.1. Integrazione e Implementazione Continua: CI/CD
 - 10.1.1. Uso di Integrazione Continua e Implementazione Continua(CI/CD)
 - 10.1.2. Differenze tra Integrazione Continua e Implementazione Continua(CI/CD)
 - 10.1.3. Integrazione e Implementazione Continua: Benefici di 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 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: Contributo
 - 10.4.2. Docker: Architettura
 - 10.4.3. Processo di implementazione 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 di CI/CD
 - 10.7.1. Integrazione continua
 - 10.7.2. Implementazione e consegna continua
 - 10.7.3. Validazioni automatiche
 - 10.7.4. Buone pratiche di CI/CD



- 10.8. CI/CD con Jenkins: Riferimenti
 - 10.8.1. CI/CD con Jenkins
 - 10.8.2. Pipeline Jenkins
 - 10.8.3. Buone pratiche con Jenkins
- 10.9. Ecosistema CD/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 CI/DC
 - 10.10.2. DevSecOps
 - 10.10.3. *Chaos Engineering*

“

*Questo approccio pratico e completo ti
preparerà ad affrontare con successo le
sfide del mercato del lavoro di oggi, dandoti
le competenze per sviluppare e mantenere
applicazioni web sofisticate e funzionali”*

07 Tirocinio

Dopo aver superato il periodo teorico online, il programma prevede un periodo di 3 settimane di formazione pratica presso un'azienda di riferimento specializzata in programmazione e sviluppo web. In questo modo, lo studente avrà a disposizione il supporto di un tutor personale, che lo accompagnerà durante tutto il processo e dedicherà il 100% del suo tempo sia alla preparazione che allo sviluppo dei tirocini.



“

I tirocini ti daranno l'opportunità di collaborare su progetti reali, sia individuali che in team, sotto la supervisione e la guida di professionisti esperti del settore"

Il periodo di formazione pratica di questo programma in Programmazione Front End Full Stack Developer è costituito da un Tirocinio in un'azienda prestigiosa, della durata di 3 settimane, dal lunedì al venerdì e con giornate di 8 ore consecutive di formazione pratica, sempre affiancato da uno specialista. Questo percorso permetterà allo studente di vedere utenti reali a fianco di un team di professionisti leader in questo ambito, applicando le procedure più innovative e pianificando la soluzione di ultima generazione per ogni problematica.

In questa proposta didattica completamente pratica, le attività sono finalizzate allo sviluppo e al perfezionamento delle competenze necessarie per l'erogazione dell'assistenza psicologica in ambiti e condizioni che richiedono un elevato livello di qualificazione, e sono orientate alla preparazione specifica per l'esercizio dell'attività, in un ambiente di sicurezza per il paziente e di elevata performance professionale. Si tratta, senza dubbio, di un'opportunità per imparare lavorando.

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

Le procedure descritte di seguito saranno la base della parte pratica della



formazione, e la loro realizzazione sarà soggetta alla disponibilità propria del centro e al suo volume di lavoro, essendo le attività proposte come segue:

Modulo	Attività Pratica
Progettazione e sviluppo di interfacce	Creare interfacce utente intuitive e coinvolgenti
	Implementare progetti responsive che si adattano a dispositivi e dimensioni diverse di schermo
	Utilizzare strumenti di progettazione e prototipazione come Adobe XD, Figma o Sketch
	Eseguire test di usabilità per garantire un'esperienza utente ottimale
Sviluppo delle applicazioni web	Sviluppare applicazioni web utilizzando tecnologie <i>front-end</i> come HTML, CSS e JavaScript
	Utilizzare <i>frameworks</i> e librerie come React, Angular o Vue.js
	Integrare API e servizi esterni per arricchire le funzionalità delle applicazioni
	Ottimizzare le prestazioni delle applicazioni web per una migliore esperienza utente
Gestione dei progetti e lavoro di squadra	Utilizzare metodologie agili come Scrum o Kanban per gestire i progetti
	Collaborare con progettisti, sviluppatori e altri membri del team
	Partecipare alle revisioni del codice e fornire un <i>feedback</i> costruttivo
	Gestire il controllo di versione utilizzando strumenti come Git e GitHub
Sviluppo back-end y database	Sviluppare e mantenere server e database utilizzando Node.js ed Express
	Progettare e gestire database relazionali e non relazionali come MySQL, PostgreSQL e MongoDB
	Implementare e garantire API RESTful per la comunicazione tra <i>front-end</i> e <i>back-end</i>
	Gestire l'autenticazione e l'autorizzazione degli utenti nelle applicazioni web

Modulo	Attività Pratica
Testing e depurazione	Scrivere ed eseguire test unitari e di integrazione utilizzando strumenti come Jest, Mocha o Jasmine
	Identificare e risolvere gli errori di codice in modo efficiente
	Automatizzare i test e le implementazioni utilizzando pipeline CI/CD
	Eseguire revisioni del codice per mantenere gli standard di qualità e le buone pratiche
Sicurezza e ottimizzazione	Implementare misure di sicurezza per proteggere le applicazioni web da minacce e vulnerabilità
	Ottimizzare le prestazioni delle applicazioni web, riducendo i tempi di caricamento e migliorando l'efficienza del codice
	Implementare tecniche di ottimizzazione SEO per migliorare la visibilità nei motori di ricerca
	Eseguire controlli di sicurezza e prestazioni delle applicazioni web



Affronterai sfide specifiche dello sviluppo web, dall'implementazione di progetti e funzionalità interattive all'integrazione di database e ottimizzazione delle prestazioni"

Assicurazione di responsabilità civile

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

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

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



Condizioni generali del tirocinio

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

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

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

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

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

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

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

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

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

08

Dove posso svolgere il Tirocinio?

Questo programma di Master Semipresenziale include un tirocinio presso un prestigioso un'azienda informatica, dove gli studenti metteranno in pratica tutto ciò che hanno imparato in materia di Programmazione Front End Full Stack Developer. In questo senso, e per avvicinare questa qualifica a un maggior numero di professionisti, TECH offre agli studenti l'opportunità di frequentarla in diversi centri in tutto il paese. In questo modo, l'istituzione rafforza il suo impegno per un'istruzione di qualità e a prezzi accessibili per tutti.



“

Oltre a rafforzare le tue capacità tecniche, i tirocini promuoveranno lo sviluppo delle tue competenze in comunicazione, gestione dei progetti e risoluzione dei problemi”

tech 46 | Dove posso svolgere il Tirocinio?



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



Informatica

Captia Ingeniería

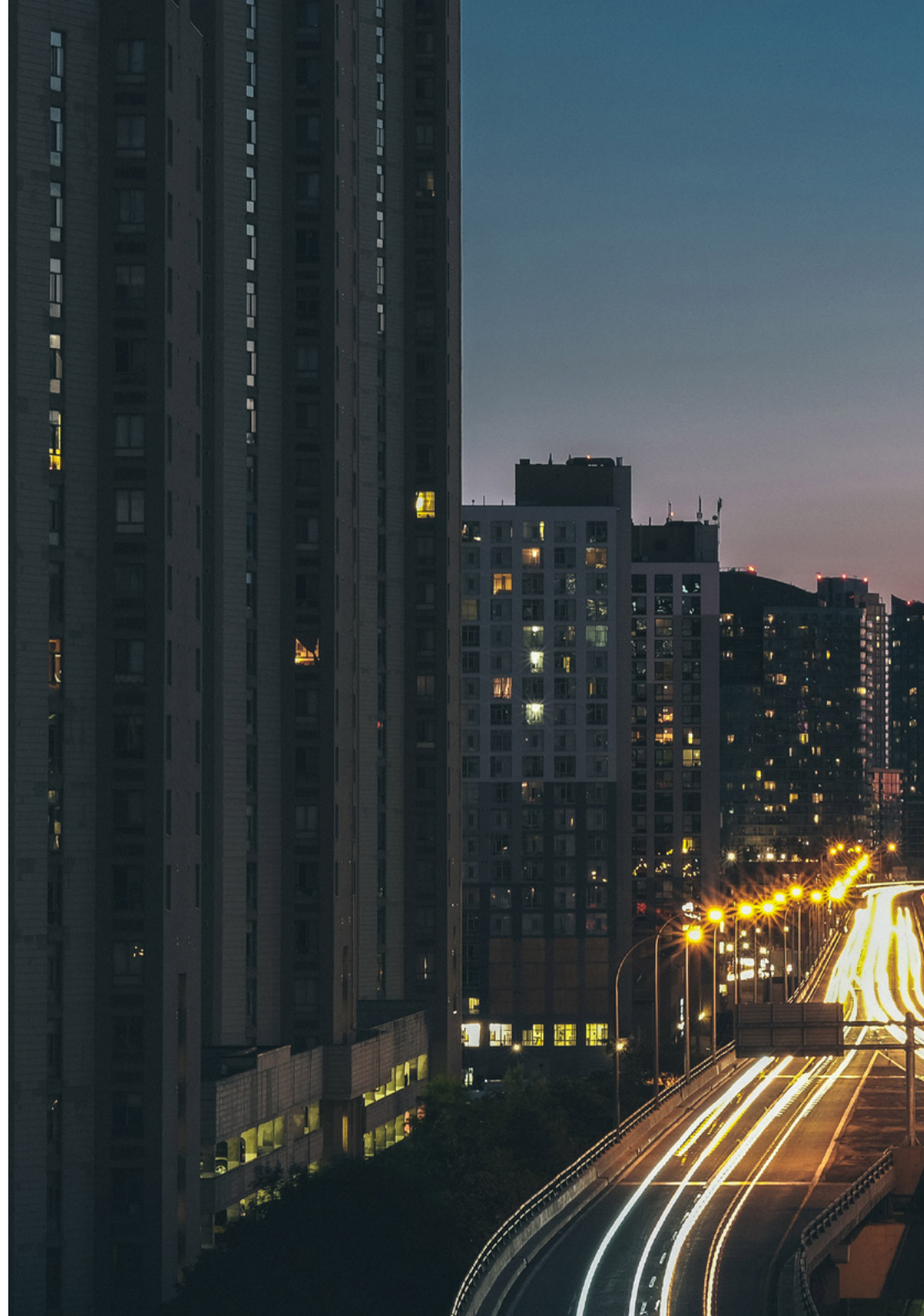
Paese: Spagna Città: Madrid

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

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

Tirocini correlati:

- Visual Analytics e Big Data
- Sviluppo di Software





“

Promuovi la tua carriera con un insegnamento olistico, che ti consente di avanzare sia a livello teorico che pratico"

09

Metodologia

Dieses Fortbildungsprogramm bietet eine andere Art des Lernens. Unsere Methodik wird durch eine zyklische Lernmethode entwickelt: **das Relearning**.

Dieses Lehrsystem wird z. B. an den renommiertesten medizinischen Fakultäten der Welt angewandt und wird von wichtigen Publikationen wie dem **New England Journal of Medicine** als eines der effektivsten angesehen.



“

Entdecken Sie Relearning, ein System, das das herkömmliche lineare Lernen hinter sich lässt und Sie durch zyklische Lehrsysteme führt: eine Art des Lernens, die sich als äußerst effektiv erwiesen hat, insbesondere in Fächern, die Auswendiglernen erfordern"

Fallstudie zur Kontextualisierung aller Inhalte

Unser Programm bietet eine revolutionäre Methode zur Entwicklung von Fähigkeiten und Kenntnissen. Unser Ziel ist es, Kompetenzen in einem sich wandelnden, wettbewerbsorientierten und sehr anspruchsvollen Umfeld zu stärken.

“

Mit TECH werden Sie eine Art des Lernens erleben, die an den Grundlagen der traditionellen Universitäten auf der ganzen Welt rüttelt”



Sie werden Zugang zu einem Lernsystem haben, das auf Wiederholung basiert, mit natürlichem und progressivem Unterricht während des gesamten Lehrplans.



Der Student wird durch gemeinschaftliche Aktivitäten und reale Fälle lernen, wie man komplexe Situationen in realen Geschäftsumgebungen löst.

Eine innovative und andersartige Lernmethode

Dieses TECH-Programm ist ein von Grund auf neu entwickeltes, intensives Lehrprogramm, das die anspruchsvollsten Herausforderungen und Entscheidungen in diesem Bereich sowohl auf nationaler als auch auf internationaler Ebene vorsieht. Dank dieser Methodik wird das persönliche und berufliche Wachstum gefördert und ein entscheidender Schritt in Richtung Erfolg gemacht. Die Fallmethode, die Technik, die diesem Inhalt zugrunde liegt, gewährleistet, dass die aktuellste wirtschaftliche, soziale und berufliche Realität berücksichtigt wird.

“ *Unser Programm bereitet Sie darauf vor, sich neuen Herausforderungen in einem unsicheren Umfeld zu stellen und in Ihrer Karriere erfolgreich zu sein*

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Informatikschulen der Welt, seit es sie gibt. Die Fallmethode wurde 1912 entwickelt, damit Jurastudenten das Recht nicht nur auf der Grundlage theoretischer Inhalte erlernen. Sie bestand darin, ihnen reale komplexe Situationen zu präsentieren, damit sie fundierte Entscheidungen treffen und Werturteile darüber fällen konnten, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard etabliert.

Was sollte eine Fachkraft in einer bestimmten Situation tun? Mit dieser Frage konfrontieren wir Sie in der Fallmethode, einer handlungsorientierten Lernmethode.

Während des gesamten Kurses werden die Studenten mit mehreren realen Fällen konfrontiert. Sie müssen ihr gesamtes Wissen integrieren, recherchieren, argumentieren und ihre Ideen und Entscheidungen verteidigen.

Relearning Methodology

TECH kombiniert die Methodik der Fallstudien effektiv mit einem 100%igen Online-Lernsystem, das auf Wiederholung basiert und in jeder Lektion verschiedene didaktische Elemente kombiniert.

Wir ergänzen die Fallstudie mit der besten 100%igen Online-Lehrmethode: Relearning.

*Im Jahr 2019 erzielten wir die besten
Lernergebnisse aller spanischsprachigen
Online-Universitäten der Welt.*

Bei TECH lernen Sie mit einer hochmodernen Methodik, die darauf ausgerichtet ist, die Führungskräfte der Zukunft zu spezialisieren. Diese Methode, die an der Spitze der weltweiten Pädagogik steht, wird Relearning genannt.

Unsere Universität ist die einzige in der spanischsprachigen Welt, die für die Anwendung dieser erfolgreichen Methode zugelassen ist. Im Jahr 2019 ist es uns gelungen, die Gesamtzufriedenheit unserer Studenten (Qualität der Lehre, Qualität der Materialien, Kursstruktur, Ziele...) in Bezug auf die Indikatoren der besten spanischsprachigen Online-Universität zu verbessern.



In unserem Programm ist das Lernen kein linearer Prozess, sondern erfolgt in einer Spirale (lernen, verlernen, vergessen und neu lernen). Daher wird jedes dieser Elemente konzentrisch kombiniert. Mit dieser Methode wurden mehr als 650.000 Hochschulabsolventen mit beispiellosem Erfolg in so unterschiedlichen Bereichen wie Biochemie, Genetik, Chirurgie, internationales Recht, Managementfähigkeiten, Sportwissenschaft, Philosophie, Recht, Ingenieurwesen, Journalismus, Geschichte, Finanzmärkte und -instrumente fortgebildet. Dies alles in einem sehr anspruchsvollen Umfeld mit einer Studentenschaft mit hohem sozioökonomischem Profil und einem Durchschnittsalter von 43,5 Jahren.

Das Relearning ermöglicht es Ihnen, mit weniger Aufwand und mehr Leistung zu lernen, sich mehr auf Ihre Spezialisierung einzulassen, einen kritischen Geist zu entwickeln, Argumente zu verteidigen und Meinungen zu kontrastieren: eine direkte Gleichung zum Erfolg.

Nach den neuesten wissenschaftlichen Erkenntnissen der Neurowissenschaften wissen wir nicht nur, wie wir Informationen, Ideen, Bilder und Erinnerungen organisieren, sondern auch, dass der Ort und der Kontext, in dem wir etwas gelernt haben, von grundlegender Bedeutung dafür sind, dass wir uns daran erinnern und es im Hippocampus speichern können, um es in unserem Langzeitgedächtnis zu behalten.

Auf diese Weise sind die verschiedenen Elemente unseres Programms im Rahmen des so genannten Neurocognitive Context-Dependent E-Learning mit dem Kontext verbunden, in dem der Teilnehmer seine berufliche Praxis entwickelt.

Dieses Programm bietet die besten Lehrmaterialien, die sorgfältig für Fachleute aufbereitet sind:



Studienmaterial

Alle didaktischen Inhalte werden von den Fachleuten, die den Kurs unterrichten werden, speziell für den Kurs erstellt, so dass die didaktische Entwicklung wirklich spezifisch und konkret ist.

Diese Inhalte werden dann auf das audiovisuelle Format angewendet, um die Online-Arbeitsmethode von TECH zu schaffen. All dies mit den neuesten Techniken, die in jedem einzelnen der Materialien, die dem Studenten zur Verfügung gestellt werden, qualitativ hochwertige Elemente bieten.



Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

Das sogenannte Learning from an Expert festigt das Wissen und das Gedächtnis und schafft Vertrauen für zukünftige schwierige Entscheidungen.



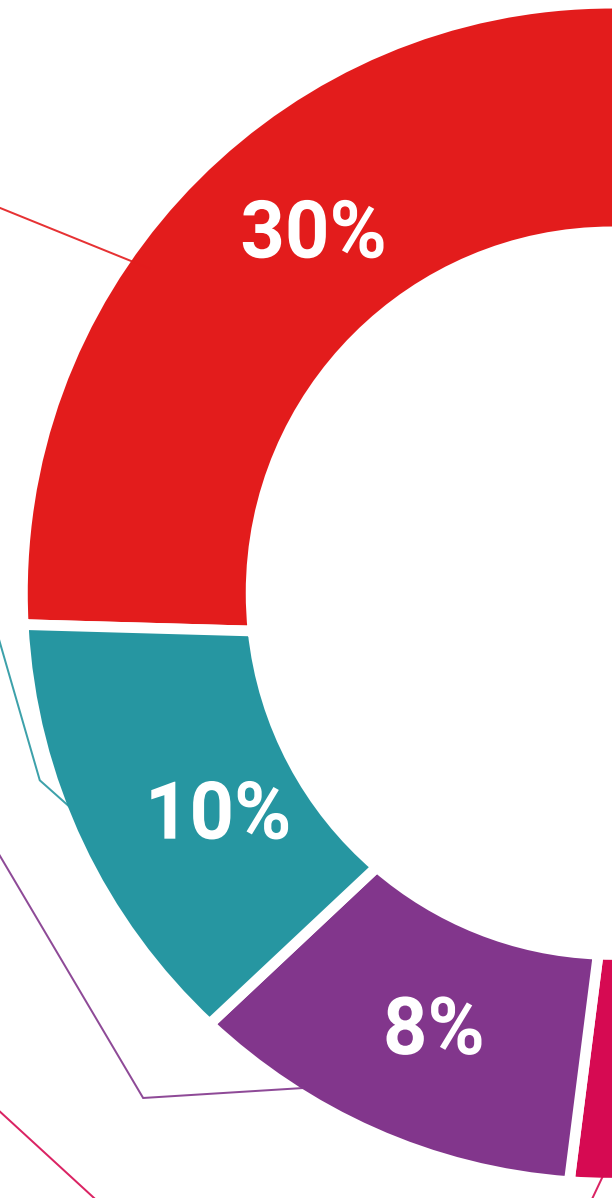
Übungen für Fertigkeiten und Kompetenzen

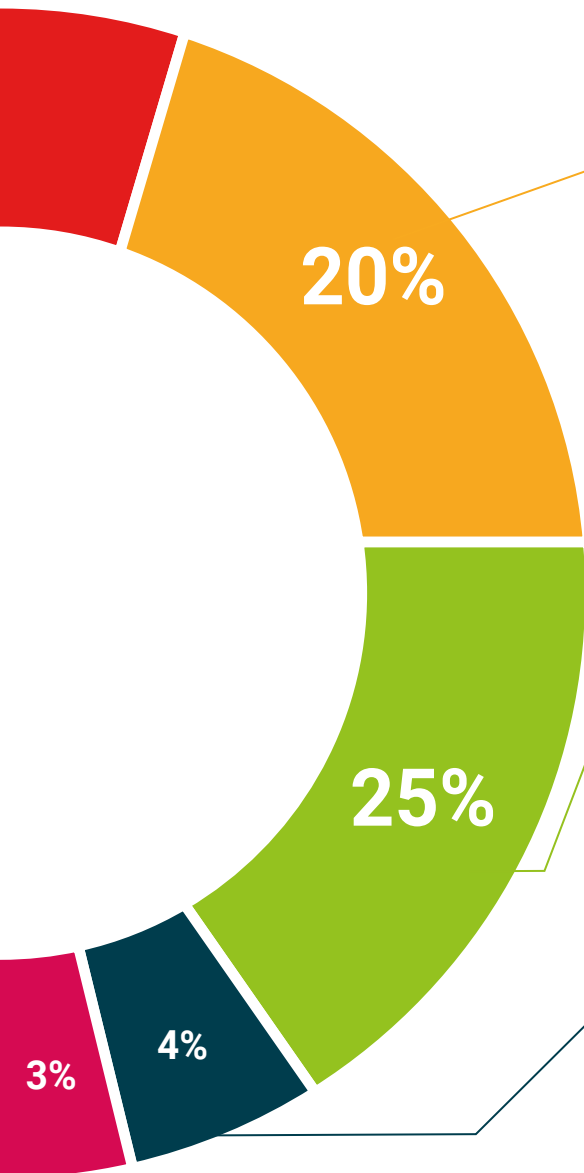
Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Übungen und Aktivitäten zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.



Weitere Lektüren

Aktuelle Artikel, Konsensdokumente und internationale Leitfäden, u. a. In der virtuellen Bibliothek von TECH hat der Student Zugang zu allem, was er für seine Fortbildung benötigt.





Case Studies

Sie werden eine Auswahl der besten Fallstudien vervollständigen, die speziell für diese Qualifizierung ausgewählt wurden. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.



Interaktive Zusammenfassungen

Das TECH-Team präsentiert die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, die Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu vertiefen.

Dieses einzigartige Bildungssystem für die Präsentation multimedialer Inhalte wurde von Microsoft als "Europäische Erfolgsgeschichte" ausgezeichnet.



Testing & Retesting

Die Kenntnisse des Studenten werden während des gesamten Programms regelmäßig durch Bewertungs- und Selbsteinschätzungsaktivitäten und -übungen beurteilt und neu bewertet, so dass der Student überprüfen kann, wie er seine Ziele erreicht.



10 Titulación

Il Master Semipresenziale in Programmazione Front End Full Stack Developer garantisce, oltre alla preparazione più rigorosa e aggiornata, il conseguimento di un qualifica di Master Semipresenziale rilasciata da TECH Global University.



“

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

Questo programma ti consentirà di ottenere il titolo di studio privato di **Master Semipresenziale in Programmazione Front End Full Stack Developer** 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.

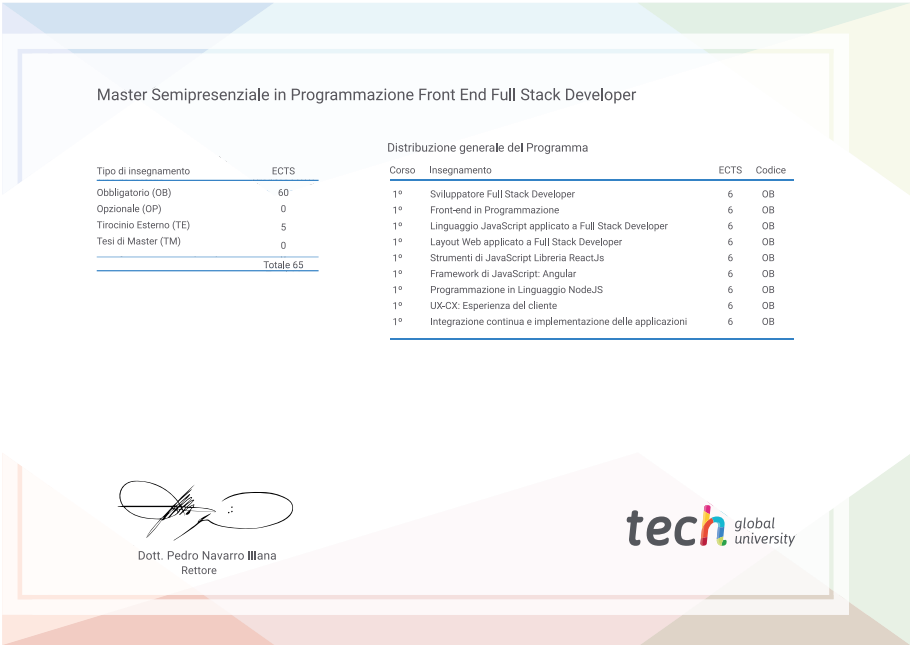
Questo titolo privato di **TECH Global University**, è un programma europeo di formazione continua e aggiornamento professionale che garantisce l'acquisizione di competenze nella propria area di conoscenza, conferendo allo studente che supera il programma un elevato valore curriculare.

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

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

Durata: **12 mesi**

Crediti: **60 + 4 ECTS**



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



Master Semipresenziale

Programmazione Front End Full Stack Developer

Modalità: Semipresenziale (Online + Tirocinio)

Durata: 12 mesi

Titolo: TECH Global University

Crediti: 60 + 4 ECTS

Master Semipresenziale

Programmazione Front End

Full Stack Developer