

Privater Masterstudiengang Front End Programmierung Full Stack Developer



Privater Masterstudiengang Front End Programmierung Full Stack Developer

- » Modalität: **online**
- » Dauer: **12 Monate**
- » Qualifizierung: **TECH Technische Universität**
- » Aufwand: **16 Std./Woche**
- » Zeitplan: **in Ihrem eigenen Tempo**
- » Prüfungen: **online**

Internetzugang: www.techtitude.com/de/informatik/masterstudiengang/masterstudiengang-front-end-programmierung-full-stack-developer

Index

01

Präsentation

Seite 4

02

Ziele

Seite 8

03

Kompetenzen

Seite 14

04

Kursleitung

Seite 18

05

Struktur und Inhalt

Seite 22

06

Methodik

Seite 34

07

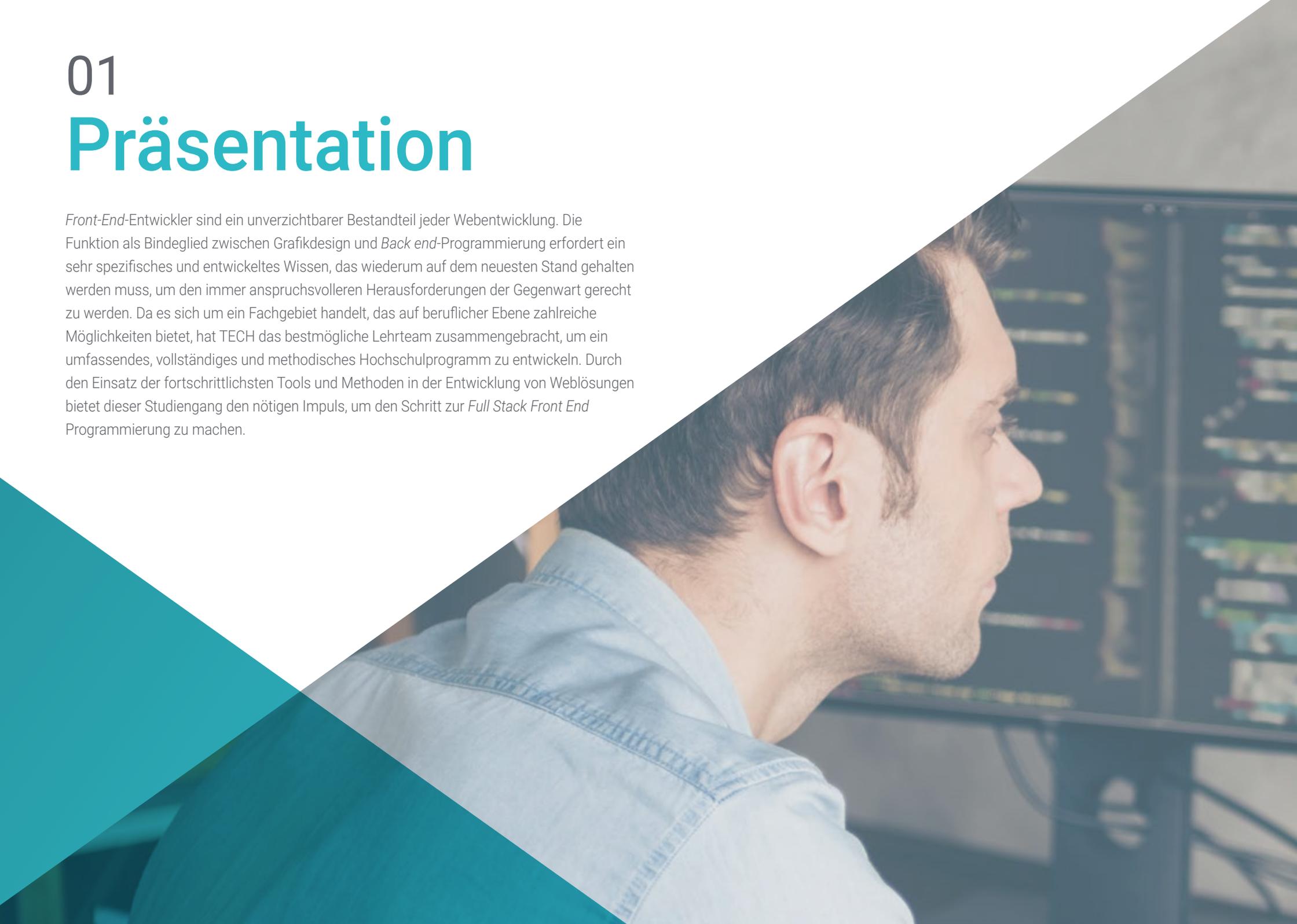
Qualifizierung

Seite 42

01

Präsentation

Front-End-Entwickler sind ein unverzichtbarer Bestandteil jeder Webentwicklung. Die Funktion als Bindeglied zwischen Grafikdesign und *Back end*-Programmierung erfordert ein sehr spezifisches und entwickeltes Wissen, das wiederum auf dem neuesten Stand gehalten werden muss, um den immer anspruchsvolleren Herausforderungen der Gegenwart gerecht zu werden. Da es sich um ein Fachgebiet handelt, das auf beruflicher Ebene zahlreiche Möglichkeiten bietet, hat TECH das bestmögliche Lehrteam zusammengebracht, um ein umfassendes, vollständiges und methodisches Hochschulprogramm zu entwickeln. Durch den Einsatz der fortschrittlichsten Tools und Methoden in der Entwicklung von Weblösungen bietet dieser Studiengang den nötigen Impuls, um den Schritt zur *Full Stack Front End* Programmierung zu machen.



“

*Werden Sie dank der perfektionierten
Arbeits- und Führungsmethoden
dieses privaten Masterstudiengangs
zu einem unverzichtbaren Bestandteil
jeder Webentwicklung, der IT-Teams
leitet und koordiniert"*

Die *Full-Stack*-Entwicklung ist eine besonders interessante Option für alle IT-Fachleute, die ihre Karriere deutlich voranbringen wollen. Die Fähigkeiten, die erforderlich sind, um in diesem Sektor gute Leistungen zu erbringen, sind umfangreich, was bedeutet, dass die Möglichkeiten, sich zu entwickeln und sogar Entwicklungsteams zu leiten, vielfältig sind.

Dank der Ausführlichkeit, mit der alle Inhalte dieses Programms entwickelt wurden, kann der Absolvent seine Karriere in Richtung *Front-End*-Webentwicklung, Seitenlayout, Spezialist für Kundenerfahrung oder DevOps ausrichten. Mit einer 360°-Sicht auf den gesamten Prozess der Erstellung eines Webs/einer Anwendung wird der Informatiker in der Lage sein, jede Art von Projekt in Angriff zu nehmen und auch die neuesten Fortschritte in allen Prozessen des Software-Lebenszyklus zu vermitteln.

Eine einzigartige akademische Möglichkeit, Zugang zu Wissen zu erhalten, das die neueste Computertheorie mit professioneller Praxis auf höchstem Niveau verbindet und von einem hochqualifizierten Lehrteam vermittelt wird. Die Erfahrung als Leiter zahlreicher einschlägiger Projekte im Bereich des digitalen Bankwesens oder der Telekommunikation bereichert den didaktischen Inhalt durch eine Vielzahl von realen Fällen und ergänzenden Lektüren.

Die große Flexibilität dieser Fortbildung ist ein weiteres herausragendes Merkmal. Es gibt keine festen Stundenpläne oder Präsenzveranstaltungen, und der Student entscheidet, wann, wo und wie er das gesamte Studienprogramm absolvieren möchte. Der gesamte Inhalt des virtuellen Klassenzimmers steht zum Download zur Verfügung und kann von jedem Gerät mit Internetanschluss aus studiert werden.

Dieser **Privater Masterstudiengang in Front End Programmierung Full Stack Developer** enthält das vollständigste und aktuellste Bildungsprogramm auf dem Markt. Seine hervorstechendsten Merkmale sind:

- ♦ Die Entwicklung von Fallstudien, die von Experten in Front End Programmierung Full Stack Developer präsentiert werden
- ♦ Der anschauliche, schematische und äußerst praxisnahe Inhalt soll wissenschaftliche und praktische Informationen zu den für die berufliche Praxis wesentlichen Disziplinen vermitteln
- ♦ Er enthält praktische Übungen in denen der Selbstbewertungsprozess durchgeführt werden kann um das Lernen zu verbessern
- ♦ Sein besonderer Schwerpunkt liegt auf innovativen Methoden
- ♦ Theoretischer Unterricht, Fragen an den Experten und individuelle Reflexionsarbeit
- ♦ Die Verfügbarkeit des Zugangs zu Inhalten von jedem festen oder tragbaren Gerät mit Internetanschluss



Schreiben Sie sich jetzt ein und verpassen Sie nicht die Gelegenheit, die Sie auf den Zenit der Leitung und Entwicklung der ehrgeizigsten IT-Projekte bringen wird"

“

Sie werden ein fortgeschrittenes Niveau an Fachwissen erreichen und in der Lage sein, jede benötigte Weblösung mit einer modernen Customer Experience-Perspektive und angepasst an den aktuellen Markt zu erstellen"

Das Dozententeam des Programms besteht aus Fachleuten des Sektors, die ihre Berufserfahrung in diese Fortbildung einbringen, sowie aus renommierten Fachleuten aus führenden Unternehmen und angesehenen Universitäten.

Die multimedialen Inhalte, die mit den neuesten Bildungstechnologien entwickelt wurden, ermöglichen den Fachleuten ein situiertes und kontextbezogenes Lernen, d. h. eine simulierte Umgebung, die ein immersives Training ermöglicht, das auf reale Situationen ausgerichtet ist.

Das Konzept dieses Studiengangs konzentriert sich auf problemorientiertes Lernen, bei dem die Fachkräfte versuchen müssen, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des akademischen Programms auftreten. Zu diesem Zweck werden sie von einem innovativen interaktiven Videosystem unterstützt, das von renommierten Experten entwickelt wurde.

Sie werden Ihr Wissen über agile Methoden und deren Umsetzung im Entwicklungsprozess vertiefen und Ihre übergreifenden Fähigkeiten und Kompetenzen erweitern.

Sie werden Zugang zu einem breiten Spektrum an Lehrmaterial haben, von der Programmiersprache Javascript bis hin zu Tools wie CSS, Angular und ReactJS.



02 Ziele

Das Ziel dieses Privaten Masterstudiengangs in Front End Programmierung Full Stack Developer ist unter Berücksichtigung der vielfältigen Möglichkeiten, die die Entwicklung bietet, die fortschrittlichsten Kenntnisse und Techniken in diesem Bereich zu vermitteln. Auf diese Weise können Informatiker sogar schon vor Abschluss des Studiums mit der Entwicklung eigener Projekte beginnen oder ihre berufliche Karriere vorantreiben, dank des eminent praktischen Ansatzes aller vermittelten Inhalte.



“

Dank des besonderen Ansatzes dieses Programms, das Sie durch alle Phasen der Full-Stack-Front-End-Entwicklung führt, werden Sie Ihre ehrgeizigsten beruflichen Ziele erreichen"



Allgemeine Ziele

- ◆ Erarbeitung von Fachwissen zu wichtigen Aspekten der Programmierung
- ◆ Ermutigung zum algorithmischen Denken
- ◆ Bereitstellung der erforderlichen Instrumente und Fähigkeiten für die Entwicklung
- ◆ Förderung der Einführung agiler Methoden für die Projektdurchführung.
- ◆ Entwicklung von Fachwissen über die Grundlagen des Internets
- ◆ Förderung des Einsatzes moderner *Front-End* Entwicklungstools und -techniken
- ◆ Entwicklung eines Webdesigns, um das Layout richtig zu gestalten
- ◆ Bewertung der erworbenen Kenntnisse





Spezifische Ziele

Modul 1. Entwicklung *Full Stack Developer*

- ◆ Entwicklung fortgeschrittener Programmierkenntnisse
- ◆ Förderung der Verwendung von Versionskontrollsystemen und Code-Hosting-Plattformen
- ◆ Förderung des Einsatzes agiler Methoden
- ◆ Vertiefung der wichtigsten Konzepte und der Funktionsweise des Internets
- ◆ Verbesserung der Befehlszeilenkenntnisse

Modul 2. *Front-End-Programmierung*

- ◆ Erkennen und Verstehen der korrekten Syntax von HTML und CSS
- ◆ Die verschiedenen Elemente von HTML erkunden
- ◆ Bestimmen des adaptiven Designansatzes
- ◆ Verwendung von Präsentationsformatierungen für Webseiten durch Anwendung von Cascading Style Sheets
- ◆ CSS-Präprozessor einbetten
- ◆ Ermitteln der Vorteile der Verwendung eines Präprozessors
- ◆ Fachwissen über Entwurfssysteme generieren
- ◆ Kriterien für die Verwendung des Entwurfssystems festzulegen

Modul 3. *JavaScript-Programmiersprache angewendet auf Full Stack Developer*

- ◆ Festlegung der von JavaScript angebotenen grundlegenden und komplexen Typen
- ◆ Die verschiedenen Möglichkeiten der Programmierung mit der Sprache zu analysieren und in jeder Situation die richtige Anwendung zu wählen
- ◆ Aktualisierung der Kenntnisse auf die neuesten Versionen
- ◆ Die funktionale Programmierung entdecken
- ◆ Untersuchung der asynchronen Programmierung und ihrer Merkmale

Modul 4. *Web-Layout angewandt auf Full Stack Developer*

- ◆ Bewertung eines Webdesigns, um zu wissen, wie es vorübergehend zu platzieren ist
- ◆ Untersuchung der wichtigsten CSS-Regeln
- ◆ Vorstellung verschiedener CSS-Methoden zur Erstellung von *Responsive Designs*
- ◆ Grundlagen der CSS-Kaskaden-Entwicklungsprinzipien
- ◆ Erkennen der Bootstrap-Technologie in jedem Webdesign
- ◆ Analyse der Grundsätze von Bootstrap
- ◆ Entwicklung eines Web-Mockups mit Bootstrap
- ◆ Bestimmung der Entwicklungsprinzipien in einem SaSS-Projekt

Modul 5. *JavaScript-Tools. ReactJs-Bibliothek*

- ◆ Bestimmung der Funktionalitäten von React
- ◆ Einrichten eines Projekts mit Create React App
- ◆ Analyse des Lebenszyklus von Komponenten in React
- ◆ Erwerb von Fachwissen über moderne React-Funktionalitäten wie *Hooks* und *Context*
- ◆ Globale Zustände mit *Context* setzen
- ◆ Erstellen und Rendern von Listen und Erstellen von Formularen mit React
- ◆ Implementierung der Feldüberprüfung in Formularen
- ◆ Styling-Komponenten und -Elemente
- ◆ Debuggen, Testen und Bereitstellen von React-Anwendungen

Modul 6. JavaScript-Framework. Angular

- ◆ Entwicklung von Fachwissen über die Architektur des Frameworks
- ◆ Vertiefung in die Angular-Methodik
- ◆ Analyse des Konzepts der Komponenten
- ◆ Den Code richtig organisieren

Modul 7. Programmierung in der Sprache NodeJS

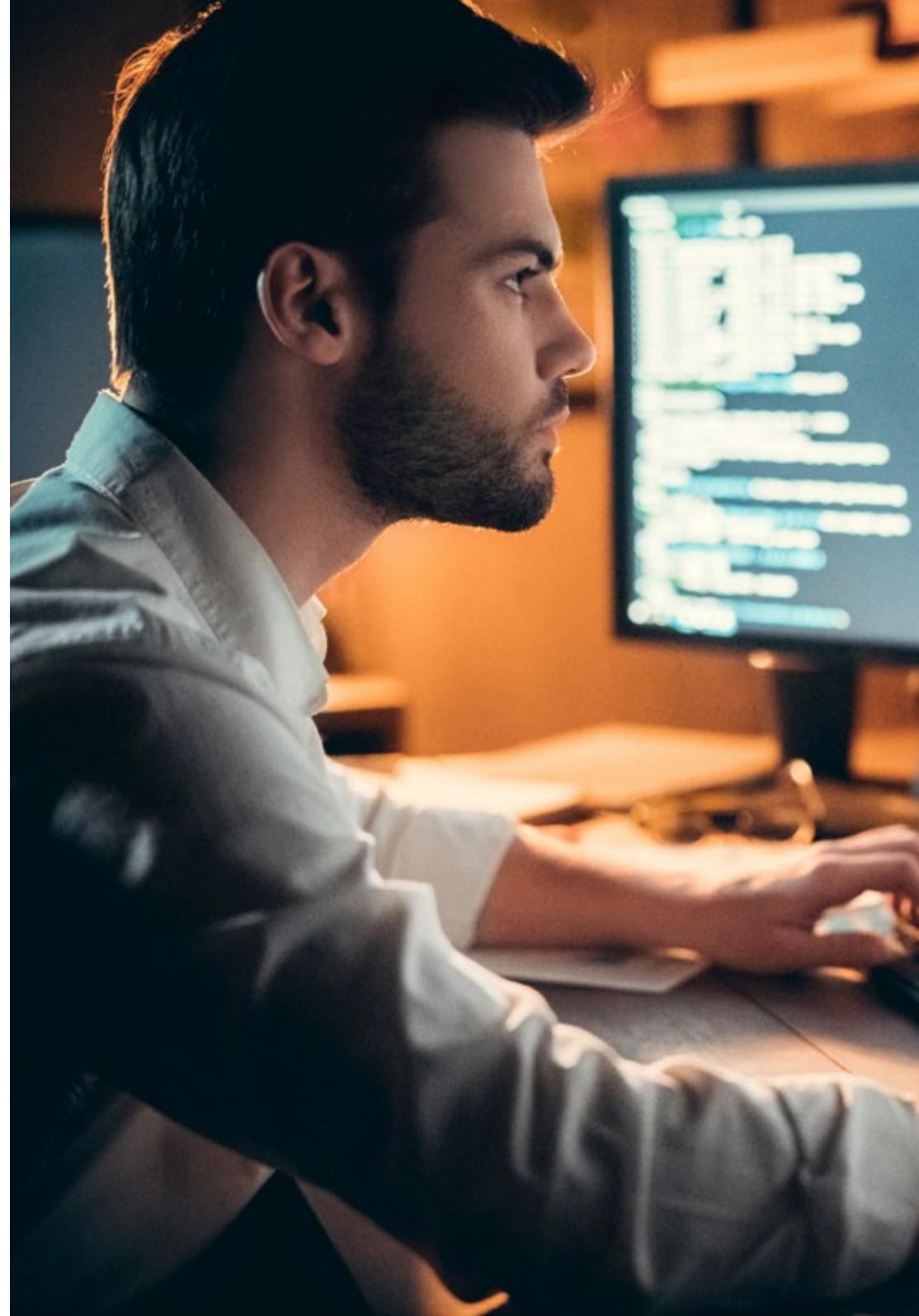
- ◆ Spezialwissen über JavaScript-Typen und ihre Operatoren generieren
- ◆ Analyse der besten Möglichkeiten zur Programmierung mit der Sprache
- ◆ Aktualisierung der Kenntnisse auf die neuesten Versionen
- ◆ Erforschung der funktionalen Programmierung
- ◆ Entwicklung der asynchronen Programmierung und ihre Motivation
- ◆ Erwerb der Fähigkeit, eine Anwendung mit NodeJSIndex zu erstellen

Modul 8. Datenbanken für Full-Stack-Entwickler

- ◆ Bestimmung der Gründe für den Einsatz einer Datenbank bei der Anwendungsentwicklung
- ◆ Untersuchung der verfügbaren Datenbanktypen und ihre Unterschiede
- ◆ Entwicklung einer klaren Vorstellung davon, wofür jede Art von Datenbank verwendet wird
- ◆ Analyse der Verwendung von Datenbanken in aktuellen Entwicklungsparadigmen

Modul 9. UX CX. Kundenerfahrung

- ◆ Analyse der Bedeutung der Nutzer heutzutage und vertiefen Sie die Feedback-Kultur
- ◆ Umsetzung von Omnichannel- und Personalisierungsstrategien auf der Grundlage von Mikrointeraktionen
- ◆ Studium der Entwicklung der Webanalyse bis hin zur Verhaltensanalytik
- ◆ Bestimmen, wie Künstliche Intelligenz CX auf die nächste Stufe gehoben hat





- ◆ Einführung der wichtigsten Analyseverfahren für Web-Erfahrung, Mobilität und Zugänglichkeit
- ◆ Vorstellung der *Design Thinking*-Methodik und des Prozesses zur Schaffung von Benutzererfahrungen
- ◆ Vorstellung konkreter Prototyping- und *WireFraming*-Tools sowie von *Frameworks* für die *Front-End*-Entwicklung

Modul 10. Kontinuierliche Integration und Anwendungsbereitstellung

- ◆ Erkennen der Vorteile eines automatisierten Anwendungsbereitstellungsmodells
- ◆ Festlegung von Unterschieden zwischen kontinuierlicher Integration, Lieferung und Bereitstellung
- ◆ Bestimmung der Hauptmerkmale von DevOps
- ◆ Bewertung einiger der grundlegenden Tools für die Implementierung von CI/CD-Pipelines
- ◆ Entwicklung der wesentlichen Faktoren für die Entwicklung von Anwendungen, die CI/CD-Prozesse unterstützen
- ◆ Untersuchung von Container-Technologien als grundlegende Säule der CI/CD-Praxis



Sie werden Ihre Fähigkeiten und Kompetenzen auf progressive Art und Weise verbessern, entlang von 10 Modulen, die auf der Grundlage des solidesten Wissens und der bewährten Erfahrung aller Dozenten erarbeitet wurden"

03

Kompetenzen

Die Fähigkeiten, die ein *Front End Full Stack* Programmierer entwickeln muss, sind vielfältig, insbesondere in einem so stark umkämpften und spezialisierten Bereich wie der IT. Aus diesem Grund deckt der Lehrplan die verschiedenen in diesem Bereich gebräuchlichen Sprachtypen sowie die Werkzeuge und die Arbeitsphilosophie ab, die der Informatiker befolgen muss, um sich beruflich zu profilieren. Dies ist dank des multidisziplinären Charakters des Lehrteams möglich, das sein gesamtes Wissen über die verschiedenen Bereiche der *Front-End*-Entwicklung gebündelt hat.



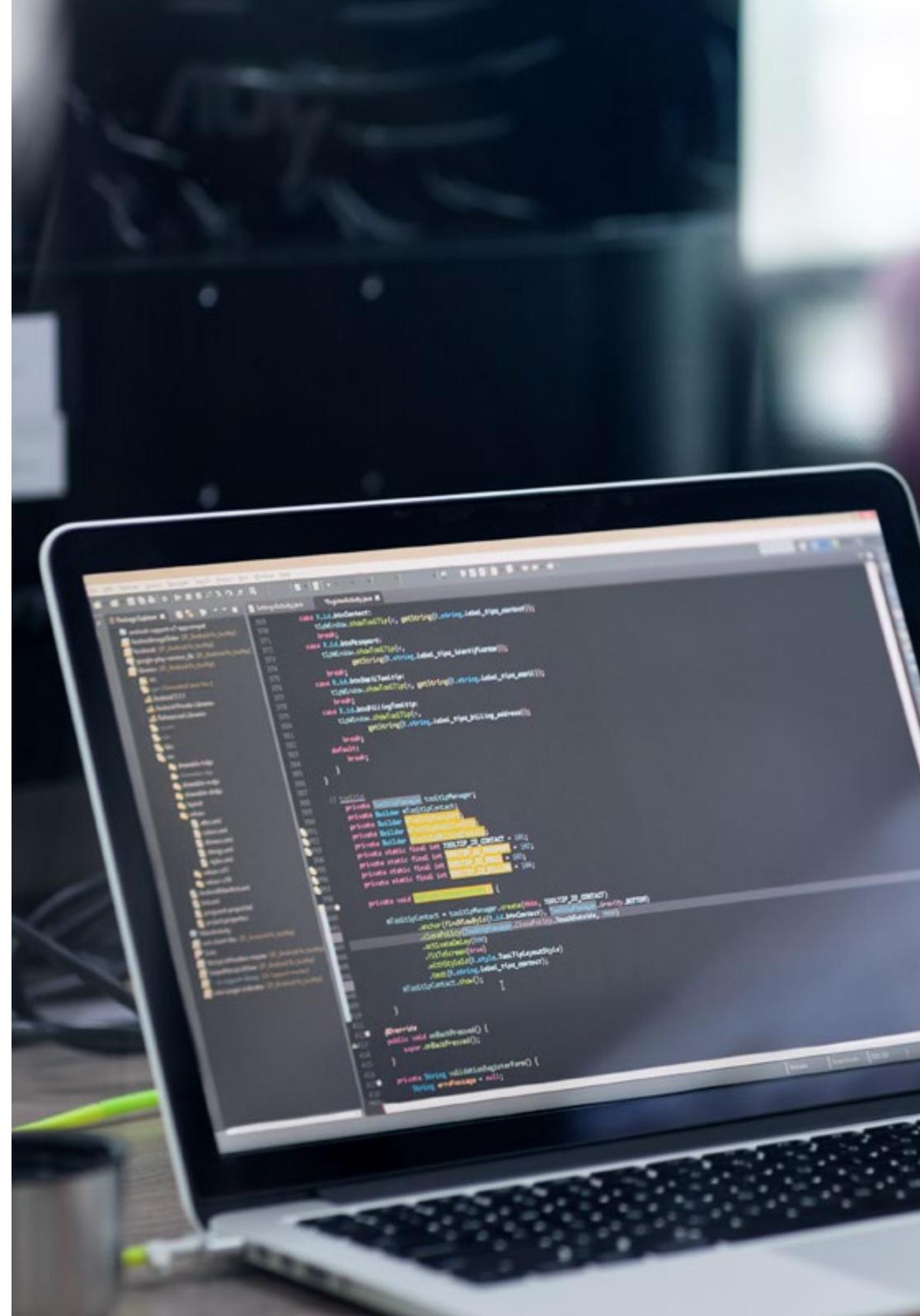
“

Sie werden Ihrem Lebenslauf einen hochwertigen Schub geben, indem Sie eine Vielzahl von Fähigkeiten und Kompetenzen einbringen, die in den höheren Ebenen der Softwareentwicklung benötigt werden"



Allgemeine Kompetenzen

- ♦ Die Syntax der Sprachen HTML und CSS richtig erkennen
- ♦ Entwicklung von Best-Practice-Kriterien für die Webentwicklung
- ♦ Spezialwissen über die Sprache JavaScript generieren
- ♦ In der Lage sein, jede Art von Anwendung mit JavaScript zu entwickeln
- ♦ Analyse der Bootstrap-Bibliothek
- ♦ Durchführung von Layoutprojekten mit SaSS (*Syntactically Awesome Stylesheets*)
- ♦ Identifizierung der React-Syntax und wie man damit programmiert
- ♦ Anwendung bewährter Verfahren im Sprachbereich
- ♦ Untersuchung des Lade- und Zugriffsprozesses in jedem der führenden Datenbanktypen in Ihrem Bereich
- ♦ Bewertung der wichtigsten Tools und Techniken für die CX-Analyse und des in Unternehmen häufig verwendeten "Technologie-Stack"





Spezifische Kompetenzen

- ◆ Analyse verschiedener Datenstrukturen
- ◆ Untersuchung von Techniken für den Entwurf und die Interpretation von Algorithmen
- ◆ Vorbereiten der Entwicklungsumgebung
- ◆ Klonen einer Website
- ◆ Erstellen einer Website mit Bootstrap
- ◆ Kompilieren von CSS-Code mit SaSS
- ◆ Entwickeln eines eigenen CSS- *Framework* auf Basis von Bootstrap mit SaSS
- ◆ Ein Projekt entwerfen und in die Tat umsetzen
- ◆ Einrichten einer Verbindung zu und Laden/Extrahieren von Daten aus verschiedenen Arten von Datenbanken
- ◆ Identifizierung von Praktiken, Anwendungsfällen, Technologien und Tools des CI/CD-Ökosystems, die für die Unterstützung des gesamten Prozesses unerlässlich sind



Sie werden im Laufe des Kurses die notwendigen Fähigkeiten entwickeln, um in der Front-End-Entwicklung erfolgreich zu sein, und dabei nicht nur Ihr eigenes Wissen, sondern auch Ihre übergreifenden Kompetenzen verbessern"

04 Kursleitung

TECH legt besonderen Wert auf die Auswahl der Lehrkräfte, die für ihre Studiengänge zuständig sind. Die Gründlichkeit, mit der die Teams zusammengestellt werden, garantiert nicht nur die Qualität der Inhalte selbst, sondern auch das Engagement eines Teams, das sich für die berufliche Weiterentwicklung der IT-Fachleute einsetzt. Der Student wird also Lehrmaterial vorfinden, das auf seine höchsten Ansprüche zugeschnitten ist, mit all der Unterstützung und dem zusätzlichen Material, das man von der größten akademischen Online-Institution der Welt erwartet.



“

Das Lehrpersonal steht Ihnen für alle Fragen zur Verfügung und bietet Ihnen eine individuelle, auf Ihre Bedürfnisse zugeschnittene Betreuung“

Leitung



Hr. Olalla Bonal, Martín

- ◆ Technischer *Blockchain*-Spezialist bei IBM SPGI
- ◆ Technischer Vertriebsspezialist für *Blockchain*. IBM
- ◆ Direktor für Architektur *Blocknitive*
- ◆ Techniker für digitale Elektronik
- ◆ *Blockchain Architekt* -IT *Infrastruktur Architekt* - IT-Projektmanager. Geschäftsbereiche: Software, Infrastruktur, Telekommunikation

Professoren

Hr. Calzada Martínez, Jesús

- ◆ Leitender Software-Ingenieur bei Devo
- ◆ *Full Stack* Entwickler bei Blocknitive
- ◆ *Front End* Manager bei Infinia
- ◆ *Full Stack* Entwickler bei Resem
- ◆ Java-Entwickler bei Hitec
- ◆ Hochschulabschluss in Computertechnik

Hr. Guerrero Díaz-Pintado, Arturo

- ◆ Technical Pre-Sales Engineer für das *Watson Customer Engagement-Portfolio* (Marketing- und *Customer Experience-Lösungen*) in Spanien, Portugal, Griechenland und Israel bei IBM
- ◆ FuE-Netzwerkingenieur bei Telefónica
- ◆ Berater für professionelle Dienstleistungen, in Zusammenarbeit mit führenden Organisationen in Europa, dem Nahen Osten und Lateinamerika
- ◆ Hochschulabschluss in Telekommunikationstechnik an der Universität von Alcalá und der *Danish Technical University*
- ◆ Herausragende Kooperationen mit renommierten Universitäten und Hochschulen in technologiebezogenen Themen wie Künstliche Intelligenz, *Internet of Things*, *Cloud*, *Customer Experience* und *Digital Transformation*

Hr. Pintado San Claudio, Bruno

- ◆ Entwicklungskordinator bei IDavinci
- ◆ Java-Entwickler bei der Nationalbibliothek von Spanien
- ◆ Support-Entwickler und Netzwerktechniker N1 bei Sanitas
- ◆ Techniker für Systemunterstützung bei der Stadtverwaltung von Alcobendas
- ◆ Kommunikationstechniker N1 für ADIF im Telekommunikationszentrum Atocha
- ◆ Hochschulabschluss in Telekommunikationstechnik, Fachrichtung elektronische Systeme, an der Polytechnischen Universität von Madrid
- ◆ Hochschulabschluss in Kommunikationselektronik an der Polytechnischen Universität von Madrid

Hr. Reyes Oliva, Luis

- ◆ Entwickler und Cloud-Architekt bei IBM
- ◆ Technischer Kundenmanager für die integrierten Konten von BBVA bei IBM
- ◆ *Cloud Executive Selling* bei IBM
- ◆ Cloud- und DevOps-Architekt bei IBM
- ◆ Architekt für Kundensoftware bei Telefónica
- ◆ Technischer Lösungsarchitekt für Rational
- ◆ *Software Engineering Manager* bei Borland
- ◆ Manager für Softwareentwicklung und Qualitätssicherung bei Altana Consulting
- ◆ Hochschulabschluss in Computertechnik an der Päpstlichen Universität von Salamanca, Madrid

Hr. Frias Favero, Pedro Luis

- ◆ CTO - Swearit Technologien
- ◆ COO - Schlüsselidentifikation
- ◆ Experte für *Blockchain* und dezentralisierte Anwendungen - Universität von Alcalá
- ◆ Full Stack Entwickler - Ironhack
- ◆ Wirtschaftsingenieur mit Abschluss an der Universität Yacambu

Hr. Gómez Rodríguez, Antonio

- ◆ Ingenieur für Cloud-Lösungen bei Oracle
- ◆ Projektleitung bei der Sopra-Gruppe
- ◆ Projektleitung bei Everis
- ◆ Projektleitung bei der staatlichen Gesellschaft zur Verwaltung von Kulturprogrammen. Andalusisches Ministerium für Kultur
- ◆ Analyst für Informationssysteme. Sopra Group
- ◆ Hochschulabschluss in Telekommunikationstechnik an der Polytechnischen Universität von Katalonien
- ◆ Aufbaustudiengang in Informationstechnologien und -systemen, Katalanisches Institut für Technologie
- ◆ *E-Business Master*, Wirtschaftshochschule La Salle

05

Struktur und Inhalt

In Anbetracht der Tatsache, dass der Aufwand und die Investition des Informatikers, der sich für ein Programm mit diesen Merkmalen entscheidet, nicht gering sind, setzt TECH auf eine erstklassige Lehrmethodik, die sich in Bezug auf Effizienz und Rentabilität bewährt hat. Das *Relearning* ermöglicht es dem Studenten, sich die wichtigsten Kenntnisse des Programms über den gesamten Lehrplan hinweg anzueignen, wodurch wertvolle Studienzzeit eingespart wird, die für die zahlreichen ergänzenden Materialien verwendet werden kann, die jedes vorgeschlagene Thema begleiten.



“

Erforschen Sie die verschiedenen Module und Wissensthemen anhand von Videozusammenfassungen, detaillierten Videos und Motivationsvideos, die von den Dozenten selbst erstellt wurden"

Modul 1. Entwicklung *Full Stack Developer*

- 1.1. Entwicklung *Full Stack Developer* I. Programmierung und Sprachen
 - 1.1.1. Programmierung
 - 1.1.2. Rollen-basierte Programmierung
 - 1.1.3. Sprachen und *Framework*
 - 1.1.4. Algorithmus
 - 1.1.5. Merkmale eines Algorithmus
- 1.2. Entwicklung *Full Stack Developer* II. Typologie
 - 1.2.1. Variablen und Konstanten
 - 1.2.2. Typen
 - 1.2.3. Betreiber
 - 1.2.4. Mitteilungen
 - 1.2.5. Schleifen
 - 1.2.6. Funktionen und Objekte
- 1.3. Datenstruktur in der Entwicklung
 - 1.3.1. Arten von linearen Strukturen
 - 1.3.2. Funktionale Strukturtypen
 - 1.3.3. Arten von Baumstrukturen
- 1.4. Entwurf und Interpretation von Algorithmen
 - 1.4.1. Parallelität in der Entwicklung. Aufteilen und erobern
 - 1.4.2. Gierige Algorithmen
 - 1.4.3. Dynamische Programmierung
- 1.5. Umgebung und Tools für *Full Stack Developer* orientierte Entwicklung
 - 1.5.1. Vorbereiten der Umgebung für Mac OS
 - 1.5.2. Vorbereiten der Linux-Umgebung
 - 1.5.3. Vorbereiten der Windows-Umgebung
- 1.6. Befehlszeile. Typologie und Funktionsweise
 - 1.6.1. Das Terminal
 - 1.6.2. Emulatoren
 - 1.6.3. Kommando-Interpreter
 - 1.6.4. Erste Kommandos
 - 1.6.5. Navigation
 - 1.6.6. Verwaltung von Dateien und Ordnern über die Befehlszeilenschnittstelle
 - 1.6.7. *Secure Shell*. SSH
 - 1.6.8. Fortgeschrittene Kommandos
- 1.7. Git. Software-Repository
 - 1.7.1. Git Software-Repository
 - 1.7.2. Verwendung von Git
 - 1.7.3. Software-Repositories
 - 1.7.4. Zweigstellen
 - 1.7.5. Arbeitszyklus
 - 1.7.6. Befehle
- 1.8. Hosting-Dienst für die Codeversionierung
 - 1.8.1. Hosting-Dienst für die Codeversionierung
 - 1.8.2. Lieferanten
 - 1.8.3. Repositories
- 1.9. Internet
 - 1.9.1. Internet
 - 1.9.2. In WWW verwendete Protokolle
 - 1.9.3. HTTP-Protokoll
- 1.10. *Full-Stack*-Entwicklungsmethodologien
 - 1.10.1. *Scrum*
 - 1.10.2. XP
 - 1.10.3. *Sprint-Design*

Modul 2. *Front-End-Programmierung*

- 2.1. HTML-Sprache
 - 2.1.1. HTML-Dokument
 - 2.1.2. Element *Head*
 - 2.1.3. Element *Body*
 - 2.1.4. Text
 - 2.1.5. Hyperlinks
 - 2.1.6. Bilder
 - 2.1.7. *Primer Site*
- 2.2. HTML-Sprache *Layouts*
 - 2.2.1. HTML-Sprache Elemente
 - 2.2.2. Traditionelles *Layout*
 - 2.2.3. Semantisches *Layout*
- 2.3. CSS (*Cascading Style Sheets*)
 - 2.3.1. Einbindung von CSS in ein HTML-Dokument
 - 2.3.2. Kommentare
 - 2.3.3. Selektoren
 - 2.3.4. Erweiterte Selektoren
- 2.4. CSS-Eigenschaften (*Cascading Style Sheets*)
 - 2.4.1. Farbe
 - 2.4.2. Text
 - 2.4.3. Pseudo-Klassen
 - 2.4.4. Übergänge
 - 2.4.5. Animationen
 - 2.4.6. Animation der Elemente
 - 2.4.7. Fortgeschrittene Animation
- 2.5. Box-Modell
 - 2.5.1. Höhe und Breite
 - 2.5.2. Rand
 - 2.5.3. Füllung
- 2.6. Positionierung
 - 2.6.1. Statische Positionierung
 - 2.6.2. Relative Positionierung
 - 2.6.3. Absolute Positionierung
 - 2.6.4. Feste Positionierung
 - 2.6.5. *Floats*
- 2.7. Anpassungsfähiges Design
 - 2.7.1. *Viewport*
 - 2.7.2. *Media Queries*
 - 2.7.3. CSS-Einheiten
 - 2.7.4. *Images*
 - 2.7.5. *Frameworks*
- 2.8. Modernes *Layout*
 - 2.8.1. *Flex*
 - 2.8.2. *Grid*
 - 2.8.3. *Flex vs. Grid*
- 2.9. Vorprozessor
 - 2.9.1. *Sass*
 - 2.9.2. Variablen
 - 2.9.3. *Mixins*
 - 2.9.4. Schleifen
 - 2.9.5. Funktionen
- 2.10. Design-System
 - 2.10.1. *Bootstrap*
 - 2.10.2. *Bootstrap-Raster*
 - 2.10.3. *Header* und *Footer* unserer Website
 - 2.10.4. *Formulare*
 - 2.10.5. *Cards*
 - 2.10.6. *Umgangsformen*

Modul 3. JavaScript-Programmiersprache angewendet auf *Full Stack Developer*

- 3.1. Primitive Typen und Operatoren
 - 3.1.1. JavaScript Sprache
 - 3.1.2. Zahlen und ihre Operatoren
 - 3.1.3. Textstrings und ihre Operatoren
 - 3.1.4. Boolesche Werte
 - 3.1.5. Umwandlung zwischen Typen
- 3.2. Strömungswächter und Struktur
 - 3.2.1. Ausdrücke und Sätze
 - 3.2.2. Variablen und Konstanten
 - 3.2.3. Anweisung *If*
 - 3.2.4. Anweisungen *For, While*
- 3.3. Funktionen
 - 3.3.1. Funktionen
 - 3.3.2. Parameter
 - 3.3.3. Funktionen als Parameter
 - 3.3.4. Umfang der Variablen
 - 3.3.5. *Scopes* verschachtelt
 - 3.3.6. *Hoisting*
 - 3.3.7. *Closures*
 - 3.3.8. Rekursion
- 3.4. Datenstrukturen: Objekte
 - 3.4.1. Typ *Object*
 - 3.4.2. Erstellung von Objekten
 - 3.4.3. Zugriff auf die Werte eines Objekts
 - 3.4.4. Hinzufügen oder Entfernen von Eigenschaften
 - 3.4.5. Verschachtelte Objekte
 - 3.4.6. *Destructuring* von Objekten
 - 3.4.7. Methoden des Typs *Object*
 - 3.4.8. *Spread Operator*
 - 3.4.9. Unveränderlichkeit
- 3.5. Datenstruktur: *Array*
 - 3.5.1. Datenstruktur. *Array*
 - 3.5.2. *Array*. Typologie
 - 3.5.3. *Arrays* verschachtelt
 - 3.5.4. Methoden einer *Array*
- 3.6. POO: *Prototype* und Klassen
 - 3.6.1. POO. Objektorientierte Programmierung
 - 3.6.2. Prototypen
 - 3.6.3. Klassen
 - 3.6.4. Private Daten
 - 3.6.5. Unterklassen
 - 3.6.6. *Call* und *Apply*
- 3.7. JavaScript-Typen
 - 3.7.1. *Set*
 - 3.7.2. *WeakSet*
 - 3.7.3. *Karte*
 - 3.7.4. *WeakMap*
 - 3.7.5. Reguläre Begriffe
- 3.8. JavaScript-Dienstprogramme
 - 3.8.1. *Date*
 - 3.8.2. *Math*
 - 3.8.3. *Symbol*
 - 3.8.4. JSON
- 3.9. JavaScript in *Browser*
 - 3.9.1. Einbindung von JavaScript in eine Website
 - 3.9.2. DOM
 - 3.9.3. Ereignisse
 - 3.9.4. *Storage* im Browser
- 3.10. Asynchrone Programmierung
 - 3.10.1. Asynchrone Programmierung
 - 3.10.2. *Event Loop*
 - 3.10.3. *Callbacks*
 - 3.10.4. *Promises*
 - 3.10.5. *Async/Await*

Modul 4. Web-Layout angewandt auf *Full Stack Developer*

- 4.1. CSS und Layout
 - 4.1.1. Layout mit Tabellen
 - 4.1.2. Flüssiges Design
 - 4.1.3. Die Ära des *Responsive*
 - 4.1.4. *Mobile First* vs. *Desktop First*
- 4.2. CSS und die Regeln des Webdesigns
 - 4.2.1. Selektoren
 - 4.2.2. Pseudo-Klassen
 - 4.2.3. Pseudo-Elemente
- 4.3. Layout mit CSS
 - 4.3.1. Regeln der *Box Model*
 - 4.3.2. Typografien
 - 4.3.3. Farbe
 - 4.3.4. Bilder
 - 4.3.5. Hintergründe
 - 4.3.6. Tabellen
 - 4.3.7. Formulare
 - 4.3.8. Einblenden und Ausblenden von Elementen
 - 4.3.9. CSS-Variablen
- 4.4. *Responsive-Design* und flüssiges Design
 - 4.4.1. Schwebende Elemente
 - 4.4.2. *Grid* CSS
 - 4.4.3. *Media Queries*
 - 4.4.4. *Flex Box*
- 4.5. Die CSS-Kaskade
 - 4.5.1. Priorität der CSS-Regeln
 - 4.5.2. Überschreiben von Regeln
 - 4.5.3. Klassen vs. Identifikatoren
- 4.6. SaSS
 - 4.6.1. Software als Dienstleistung (SaSS)
 - 4.6.2. Installation von SaSS
 - 4.6.3. Ausführen und Kompilieren von SaSS
 - 4.6.4. Struktur eines SaSS-Verzeichnisses
- 4.7. Verwendung von SaSS
 - 4.7.1. Variablen in Sass
 - 4.7.2. Modularisierung unseres Projekts
 - 4.7.3. SaSS-Syntax
- 4.8. SaSS-Logik
 - 4.8.1. Mixins
 - 4.8.2. Maps
 - 4.8.3. Funktionen und Kontrollstrukturen
- 4.9. Layout mit Bootstrap
 - 4.9.1. Bootstrap
 - 4.9.2. *Layout* Bootstrap
 - 4.9.3. Formulare
 - 4.9.4. *Box Model* mit Bootstrap
 - 4.9.5. Farben und Schriftarten
 - 4.9.6. Links und Schaltflächen
 - 4.9.7. Ein- und Ausblenden von Elementen mit Bootstrap
 - 4.9.8. *Flex Box* mit Bootstrap
 - 4.9.9. Komponenten
- 4.10. *Theming* Bootstrap
 - 4.10.1. Bootstrap mit SaSS umschreiben (*Software as a Service*)
 - 4.10.2. Struktur der Datei
 - 4.10.3. Erstellung eines eigenen CSS *Framework* (*Cascading Style Sheets*)

Modul 5. Javascript-Tools. ReactJs-Bibliothek

- 5.1. ReactJS Javascript Tool
 - 5.1.1. Das ReactJS-Tool
 - 5.1.2. Create React App
 - 5.1.3. JavaScript *Syntax Extension*
- 5.2. ReactJS-Komponenten
 - 5.2.1. Komponenten
 - 5.2.2. *Props*
 - 5.2.3. Rendering
- 5.3. Ereignisse in der ReactJS-Bibliothek
 - 5.3.1. Event Management
 - 5.3.2. Online-Event Management
 - 5.3.3. Ereignisse in der ReactJS-Bibliothek
- 5.4. ReactJS *Hooks* konfigurieren
 - 5.4.1. Status einer Komponente
 - 5.4.2. *Status Hook*
 - 5.4.3. *Effect Hook*
 - 5.4.4. *Custom Hooks*
 - 5.4.5. *Andere Hooks*
- 5.5. *Component Context* in ReactJS
 - 5.5.1. *Component Context* in ReactJS
 - 5.5.2. Verwendung von *Context*
 - 5.5.3. Struktur des *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. Praktische Anwendung des Einsatzes von *Context*
- 5.6. Leitweglenkung in ReactJs
 - 5.6.1. *Router*
 - 5.6.2. *React Router*
 - 5.6.3. Installation
 - 5.6.4. Basis-Routing
 - 5.6.5. Dynamisches Routing
 - 5.6.6. Primäre Komponenten
 - 5.6.7. *React Router Hooks*
- 5.7. Verwendung von Listen und Formularen mit ReactJS
 - 5.7.1. Listen und Schleifen
 - 5.7.2. Formulare und Validierungen
 - 5.7.3. *React Hook Forms*
- 5.8. Verwendung von Stilen in ReactJS
 - 5.8.1. Traditionell gestylt
 - 5.8.2. Online-Stile
 - 5.8.3. Hinzufügung einer Entwurfssystembibliothek
- 5.9. Javascript-Tests. Tools
 - 5.9.1. *Testing*
 - 5.9.2. *Jest JavaScript Testing Framework*
 - 5.9.3. *Visuelle Prüfung und Dokumentation*
- 5.10. Einsatz von Code mit ReactJS
 - 5.10.1. *Hosting*
 - 5.10.2. Lieferanten
 - 5.10.3. Projektvorbereitung
 - 5.10.4. Bereitstellung auf Heroku



Modul 6. JavaScript-Framework Angular

- 6.1. Das Angular *Framework* und seine Architektur
 - 6.1.1. Angular CLI
 - 6.1.2. Architektur
 - 6.1.3. *Workspace* und Struktur
 - 6.1.4. Umgebung
- 6.2. Komponenten des Angular- *Framework*
 - 6.2.1. Lebenszyklus
 - 6.2.2. Verkapselung der Sicht
 - 6.2.3. Interaktion zwischen den Komponenten
 - 6.2.4. Projektion des Inhalts
- 6.3. Angular *Framework* -Vorlagen
 - 6.3.1. Text-Interpolation
 - 6.3.2. Mitteilungen
 - 6.3.3. *Property Binding*
 - 6.3.4. *Class, Style und Attribute Binding*
 - 6.3.5. *Event Binding* und *Two-Way Binding*
 - 6.3.6. *Pipes*
- 6.4. Angular *Framework* Direktiven
 - 6.4.1. Angular-Richtlinien
 - 6.4.2. Richtlinien für Attribute
 - 6.4.3. Strukturelle Richtlinien
- 6.5. Dienste und Injektion von Abhängigkeiten
 - 6.5.1. Dienste
 - 6.5.2. Injektion von Abhängigkeiten
 - 6.5.3. *Service Providers*

- 6.6. *Routing* und Navigation
 - 6.6.1. Anwendung mit *Routing*
 - 6.6.2. Basis-Routing
 - 6.6.3. Verschachtelte Routen
 - 6.6.4. Parameter
 - 6.6.5. Zugang und Genehmigung
 - 6.6.6. *Lazy Loading* von Modulen
- 6.7. RxJS
 - 6.7.1. Beobachtbare Daten
 - 6.7.2. *Observers*
 - 6.7.3. Abonnements
 - 6.7.4. Betreiber
- 6.8. Formulare und HTTP
 - 6.8.1. Reaktive Formulare
 - 6.8.2. Feldvalidierung
 - 6.8.3. Dynamische Formulare
 - 6.8.4. Petitionen
 - 6.8.5. *Interceptors*
 - 6.8.6. Sicherheit
- 6.9. Animationen
 - 6.9.1. Übergänge und *Triggers*
 - 6.9.2. Routenübergänge
 - 6.9.3. Unterschiede zwischen den Übergängen
- 6.10. *Testing* im Angular *Framework*
 - 6.10.1. Testen von Diensten
 - 6.10.2. Testen von Komponenten
 - 6.10.3. Testen von Richtlinien und *pipes*

Modul 7. Programmierung in der Sprache NodeJS

- 7.1. NodeJS und seine Architektur
 - 7.1.1. NPM und Paketverwaltung
 - 7.1.2. Durchführung eines Programms
 - 7.1.3. Module
 - 7.1.4. Erstellung eines Moduls
 - 7.1.5. *Loop* der Ereignisse
- 7.2. Backend-Server, HTTP, *Express* und *Sockets*
 - 7.2.1. Modul HTTP
 - 7.2.2. *Express*
 - 7.2.3. *Socket.io*
- 7.3. Datenbanken und Zwischenspeicherung
 - 7.3.1. MongoDB
 - 7.3.2. Mongoose
 - 7.3.3. SQL
 - 7.3.4. *Sequelize*
 - 7.3.5. *Redis*
- 7.4. Dateisystem und *Os*
 - 7.4.1. *File System Module*
 - 7.4.2. *Os Module*
 - 7.4.3. *Cluster Module*
- 7.5. Veranstaltungen, *Buffers* und *Streams*
 - 7.5.1. *Events*
 - 7.5.2. *Puffer*
 - 7.5.3. *Streams*
- 7.6. *Testing*
 - 7.6.1. *Jest*
 - 7.6.2. *Mocha*
 - 7.6.3. TDD - *Cucumber*

- 7.7. Architektur und gute Praxis
 - 7.7.1. DRY
 - 7.7.2. SOLID
 - 7.7.3. CRUD
 - 7.7.4. MVC
 - 7.7.5. Monolithen
 - 7.7.6. Microservices
 - 7.7.7. Sechseckige Architekturen
- 7.8. Typescript
 - 7.8.1. Typen, Schnittstellen und Klassen
 - 7.8.2. Funktionen und Module
 - 7.8.3. Generika
 - 7.8.4. *Namespaces*
 - 7.8.5. Dekorationen
- 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. Erstellen einer REST-API mit Express
- 7.10. Erstellung und Containerisierung einer Anwendung mit NestJS
 - 7.10.1. Nest CLI
 - 7.10.2. Docker
 - 7.10.3. Aufbau einer App

Modul 8. Datenbanken für *Full Stack*-Entwickler

- 8.1. Datenbanken für *Full Stack*-Entwickler
 - 8.1.1. Datenbank in der Anwendungsentwicklung
 - 8.1.2. Datenbank-Fähigkeiten
 - 8.1.3. SQL (*Structured Query Language*)
- 8.2. Wahl der Datenbank
 - 8.2.1. Zu berücksichtigende Anwendung oder Dienstleistung
 - 8.2.2. Kategorien von Datenbanken
 - 8.2.3. Datenbank-Übersicht
- 8.3. Entwicklung mit MySQL
 - 8.3.1. Entwicklung mit MySQL
 - 8.3.2. Einsatz eines relationalen Modells mit MySQL
 - 8.3.3. MySQL-Verbindung
- 8.4. Entwicklung mit Oracle Database
 - 8.4.1. Entwicklung mit Oracle DB
 - 8.4.2. Einsatz des Modells
 - 8.4.3. Verbindung zur Oracle Database
- 8.5. Entwicklung mit Oracle SQL Server
 - 8.5.1. Oracle SQL Server
 - 8.5.2. Einsatz des Modells
 - 8.5.3. Verbindung zu SQL Server
- 8.6. NoSQL-Entwicklung
 - 8.6.1. Vergleich mit SQL-Datenbanken
 - 8.6.2. Erstellung einer Datenbank in MongoDB
 - 8.6.3. Verbindung zu MongoDB
- 8.7. Entwicklung mit grafos
 - 8.7.1. Entwicklung mit grafos
 - 8.7.2. Datenbankerstellung mit Neo4j
 - 8.7.3. Verbindung zu Neo4j

- 8.8. Entwicklung von Schlüsselwert-Datenbanken
 - 8.8.1. Entwicklung mit k-v Datenbank
 - 8.8.2. Datenbankerstellung mit Redis
 - 8.8.3. Verbindung zu Redis
- 8.9. Datenbanken mit anderen Datentypen
 - 8.9.1. *Elastic Search*
 - 8.9.2. *Inmemory Database*
 - 8.9.3. Entwicklung mit räumlichen Daten
- 8.10. Datenbanken. Fortgeschrittene Aspekte
 - 8.10.1. Datenbanken in nativen Cloud-Entwicklungen
 - 8.10.2. Datenbanken in der Microservices-Architektur
 - 8.10.3. CI/CD und Datenbanken

Modul 9. UX CX. Kundenerfahrung

- 9.1. *Customer Experience*
 - 9.1.1. *Customer Experience (CX)*
 - 9.1.2. Neue Bedürfnisse der Verbraucher
 - 9.1.3. *Feedback in Customer Experience*
- 9.2. Innovative Technologien
 - 9.2.1. Maschinen, die denken
 - 9.2.2. Neue Wege der Informationsweitergabe
 - 9.2.3. Messen, was nicht gemessen werden kann
- 9.3. Kanäle für die Benutzerinteraktion
 - 9.3.1. Kundenanalyse
 - 9.3.2. Personalisierung
 - 9.3.3. Mehrere Kanäle der Benutzerinteraktion

- 9.4. Benutzer-Analytik
 - 9.4.1. Struktur der Website
 - 9.4.2. Benutzer-Analytik
 - 9.4.3. Erweiterte Benutzeranalyse
- 9.5. Nielsen und sein Einfluss auf die CX
 - 9.5.1. Nielsen und sein Einfluss auf die CX
 - 9.5.2. Techniken für *User Testing*
- 9.6. Tools für *Customer Experience*
 - 9.6.1. Erweiterte Tools
 - 9.6.2. Mobilität
 - 9.6.3. Zugänglichkeit
- 9.7. Neue Methodologien
 - 9.7.1. Die Herausforderung für den Nutzer
 - 9.7.2. UX-Prozess
 - 9.7.3. Benutzerforschung
- 9.8. Kommunikation eines Entwurfs
 - 9.8.1. *Wireframing*
 - 9.8.2. Kommunikationsmittel eines Entwurfs
 - 9.8.3. Fortgeschrittene Werkzeuge für die Designkommunikation
- 9.9. UI-Entwurf
 - 9.9.1. UI-Entwurf
 - 9.9.2. Web- und mobile Schnittstellen
 - 9.9.3. Web- und mobile Komponenten
- 9.10. Erstellung einer CX
 - 9.10.1. Erstellung einer CX
 - 9.10.2. Neue Erlebnisse entwerfen
 - 9.10.3. Schnittstellen

Modul 10. Kontinuierliche Integration und Anwendungsbereitstellung

- 10.1. Kontinuierliche Integration und kontinuierliche Bereitstellung: CI/CD
 - 10.1.1. Einsatz von kontinuierlicher Integration und kontinuierlicher Bereitstellung (CI/CD)
 - 10.1.2. Unterschiede zwischen Continuous Integration und Continuous Delivery (CI/CD)
 - 10.1.3. Kontinuierliche Integration und kontinuierliche Bereitstellung. Vorteile von CI/CD
- 10.2. Neue Entwicklungsparadigmen
 - 10.2.1. Vom Waterfall zum DevOps
 - 10.2.2. Style Guide: die 12 Faktoren
 - 10.2.3. Cloud Native, Microservices und *Serverless*
- 10.3. DevOps, über CI/CD hinaus
 - 10.3.1. DevOps
 - 10.3.2. DevOps. *Continuous Everything*
 - 10.3.3. DevOps vs SRE
- 10.4. Container-Technologie I - Docker
 - 10.4.1. Container. Beitrag
 - 10.4.2. Docker. Architektur
 - 10.4.3. Bereitstellungsprozess mit Docker
- 10.5. Container-Technologie II - Kubernetes
 - 10.5.1. Orchestrierung
 - 10.5.2. Kubernetes
 - 10.5.3. Das Kubernetes-Ökosystem
- 10.6. Konfiguration der Infrastruktur mit GitOps
 - 10.6.1. Unveränderliche Infrastruktur
 - 10.6.2. GitOps
 - 10.6.3. GitOps-Tools
- 10.7. Pipelines und Automatisierung. CI/CD Anwendungsfälle
 - 10.7.1. Kontinuierliche Integration
 - 10.7.2. Bereitstellung und kontinuierliche Lieferung
 - 10.7.3. Automatische Überprüfungen
 - 10.7.4. Gute Praxis bei CI/CD
- 10.8. CI/CD mit Jenkins. Referenzen
 - 10.8.1. CI/CD mit Jenkins
 - 10.8.2. Jenkins-Pipelines
 - 10.8.3. Gute Praxis mit Jenkins
- 10.9. CI/CD-Ökosystem
 - 10.9.1. Organisation des Ökosystems
 - 10.9.2. Erweiterte Tools
 - 10.9.3. Dagger. Die Zukunft
- 10.10. Letzte Phasen des CI/CD-orientierten Softwarezyklus
 - 10.10.1. Anwendung von KI auf den IC/DC-Prozess
 - 10.10.2. DevSecOps
 - 10.10.3. *Chaos Engineering*



Ihnen werden die modernsten Bildungsressourcen zur Verfügung stehen, mit kostenlosem Zugang zum virtuellen Klassenzimmer 24 Stunden am Tag"

06 Methodik

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 aufgibt 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 die Grundlagen der traditionellen Universitäten in der ganzen Welt verschiebt”



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



Die Studenten lernen durch gemeinschaftliche Aktivitäten und reale Fälle die Lösung komplexer Situationen in realen Geschäftsumgebungen.

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 die Jurastudenten das Recht nicht nur anhand theoretischer Inhalte erlernen, sondern ihnen reale, komplexe Situationen vorlegen, damit sie fundierte Entscheidungen treffen und Werturteile darüber fällen können, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard eingeführt.

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 Studierenden mit mehreren realen Fällen konfrontiert. Sie müssen Ihr gesamtes Wissen integrieren, recherchieren, argumentieren und Ihre Ideen und Entscheidungen verteidigen.

Relearning Methodik

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 auszubilden. 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 Online-Universität in Spanisch 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 ausgebildet. 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 Ihr Fachgebiet 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 neurokognitiven kontextabhängigen E-Learnings 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 TECH-Online-Arbeitsmethode zu schaffen. Und das alles mit den neuesten Techniken, die dem Studenten qualitativ hochwertige Stücke aus jedem einzelnen Material zur Verfügung stellen.



Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

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



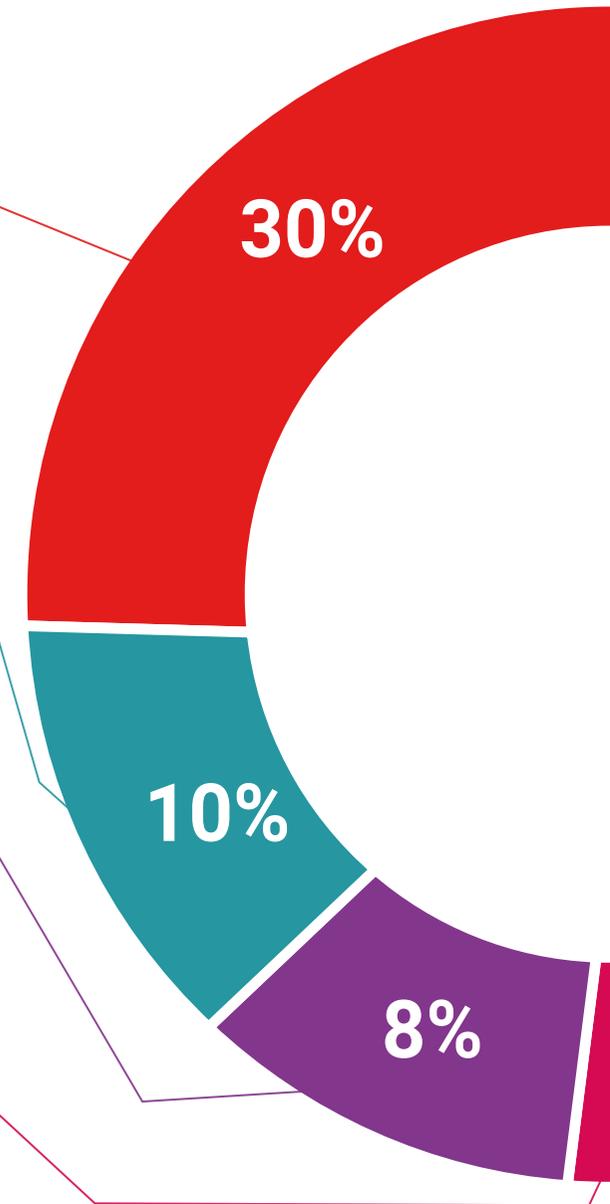
Fertigkeiten und Kompetenzen Praktiken

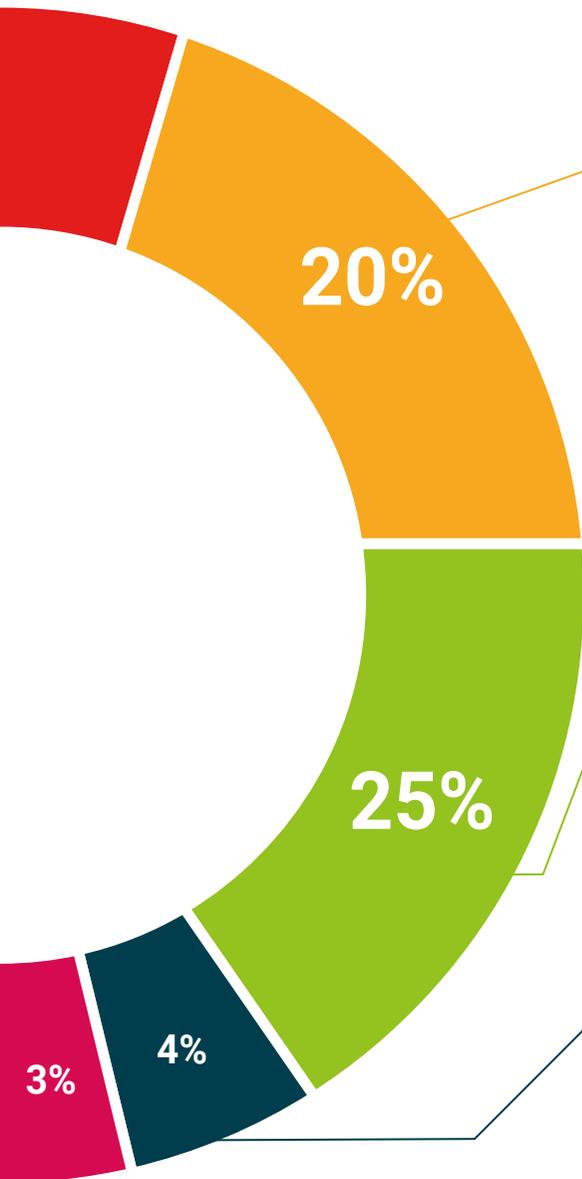
Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Praktiken und Dynamiken zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.



Weitere Lektüren

Aktuelle Artikel, Konsensdokumente und internationale Leitfäden, u.a. In der virtuellen Bibliothek von TECH haben die Studenten Zugang zu allem, was sie für ihre Ausbildung benötigen.





Fallstudien

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.



Prüfung und Nachprüfung

Die Kenntnisse der Studenten werden während des gesamten Programms regelmäßig durch Bewertungs- und Selbsteinschätzungsaktivitäten und -übungen beurteilt und neu bewertet, so dass die Studenten überprüfen können, wie sie ihre Ziele erreichen.



07

Qualifizierung

Der Privater Masterstudiengang in Front End Programmierung Full Stack Developer garantiert neben der strengsten und aktuellsten Ausbildung auch den Zugang zu einem von der TECH Technologischen Universität ausgestellten Diplom.



“

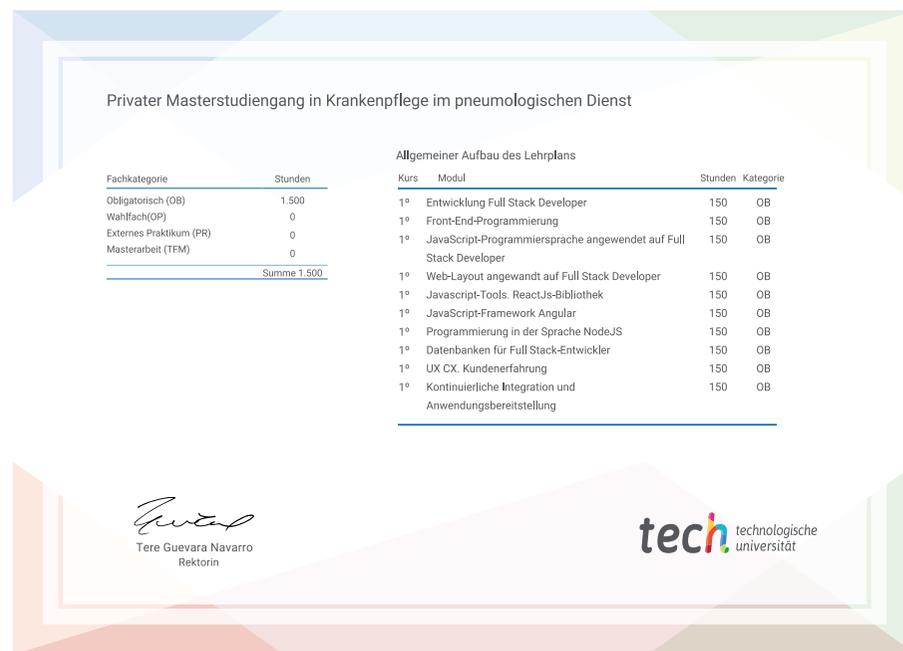
*Schließen Sie dieses Programm erfolgreich ab
und erhalten Sie Ihren Universitätsabschluss
ohne lästige Reisen oder Formalitäten"*

Dieser **Privater Masterstudiengang in Front End Programmierung Full Stack Developer** enthält das vollständigste und aktuellste Programm auf dem Markt.

Sobald der Student die Prüfungen bestanden hat, erhält er/sie per Post* mit Empfangsbestätigung das entsprechende Diplom, ausgestellt von der **TECH Technologischen Universität**.

Das von **TECH Technologische Universität** ausgestellte Diplom drückt die erworbene Qualifikation aus und entspricht den Anforderungen, die in der Regel von Stellenbörsen, Auswahlprüfungen und Berufsbildungsausschüssen verlangt werden.

Titel: **Privater Masterstudiengang in Front End Programmierung Full Stack Developer**
Anzahl der offiziellen Arbeitsstunden: **1.500 Std.**



*Haager Apostille. Für den Fall, dass der Student die Haager Apostille für sein Papierdiplom beantragt, wird TECH EDUCATION die notwendigen Vorkehrungen treffen, um diese gegen eine zusätzliche Gebühr zu beschaffen.

zukunft

gesundheit vertrauen menschen
erziehung information tutoren
garantie akkreditierung unterricht
institutionen technologie lernen
gemeinschaft verpflichtung
persönliche betreuung innovation
wissen gegenwart qualität
online-Ausbildung
entwicklung instituten
virtuelles Klassenzimmer

tech technologische
universität

Privater Masterstudiengang
Front End Programmierung
Full Stack Developer

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technologische Universität
- » Aufwand: 16 Std./Woche
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Privater Masterstudiengang Front End Programmierung Full Stack Developer