

Privater Masterstudiengang Programmierung in der Cloud





Privater Masterstudiengang Programmierung in der Cloud

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

Internetzugang: www.techtitute.com/de/informatik/masterstudiengang/masterstudiengang-programmierung-cloud

Index

01

Präsentation

Seite 4

02

Ziele

Seite 8

03

Kompetenzen

Seite 14

04

Kursleitung

Seite 18

05

Struktur und Inhalt

Seite 24

06

Methodik

Seite 36

07

Qualifizierung

Seite 44

01 Präsentation

Unternehmen und *Start-ups* haben mit der digitalen Transformation begonnen, bei der *Cloud*-Umgebungen ihre Arbeitsprozesse rationalisieren und ihnen mehr Sicherheit und Kostenkontrolle bieten: ein technologischer Umbruch, der in vollem Gange ist. Unternehmen benötigen Fachkräfte, die in der Lage sind, *Clouds* zu konzipieren, aufzubauen, zu konfigurieren und zu verwalten. Diese Qualifikation bietet die Möglichkeit, sich zu spezialisieren und Kenntnisse über *Cloud*-Technologien zu erwerben. Der Online-Modus dieses Kurses und das *Relearning*-System, das auf der Wiederholung von Inhalten basiert, werden das Studium erleichtern, insbesondere für diejenigen, die Flexibilität und Autonomie bei der Verteilung des Lernpensums dieses Kurses wünschen.



“

Dank dieses privaten Masterstudiengangs werden Sie in der Lage sein, relevante Positionen im wettbewerbsintensiven Cloud Computing-Sektor zu besetzen”

Softwareentwicklung und *Cloud*-Umgebungen haben zur Entstehung einer neuen Berufsgruppe im Bereich der neuen Technologien geführt. In kurzer Zeit haben die Unternehmen die Vorteile erkannt, die ihnen die Nutzung der *Cloud Computing*-Technologie bietet. Angesichts dieses Szenarios haben IT-Fachkräfte die Möglichkeit, sich in einem aufstrebenden Bereich weiterzuentwickeln.

Dieser private Masterstudiengang von TECH vereint ein Dozententeam, das sich auf *Cloud*-Technologien spezialisiert hat und über umfassende Erfahrung in diesem Sektor verfügt. Ihr Wissen gibt den Studenten alle notwendigen Werkzeuge an die Hand, um die verschiedenen *Cloud*-Anbieter kennenzulernen und so alle Technologien zu beherrschen, die von den großen Anbietern von *Cloud*-Lösungen angeboten werden. Darüber hinaus werden IT-Profis unter der Leitung von Experten in diesem Bereich die wichtigsten Konzepte und Werkzeuge kennen lernen, die derzeit im Bereich der Datenpersistenz verwendet werden, wie beispielsweise *Data Lakes*.

Während der zwölfmonatigen Dauer des Kurses werden die Themen Virtualisierung und Containerisierung von Anwendungen behandelt, die die Entwicklung des Systemmanagements vorangetrieben haben und heute von grundlegender Bedeutung sind. All dies aus einer theoretischen und praktischen Perspektive, entwickelt von und für zukünftige *Cloud*-Architekten, DevOps oder *Cloud*-Infrastruktur-Spezialisten.

Eine hervorragende Gelegenheit für Fachleute, die ihre beruflichen Ambitionen durch diesen 100%igen Online-Kurs verbessern möchten. Sie benötigen lediglich ein internetfähiges Gerät, um auf die Bibliothek mit Multimedia-Ressourcen und praktischen Simulationen zuzugreifen, die das Studium erleichtern und ihnen die Flexibilität geben, es mit ihren anspruchsvollen beruflichen und privaten Verpflichtungen zu vereinbaren.

Dieser **Privater Masterstudiengang in Programmierung in der Cloud** enthält das vollständigste und aktuellste Programm auf dem Markt. Die hervorstechendsten Merkmale sind:

- ◆ Die Entwicklung von Fallstudien, die von Experten für Programmierung in der *Cloud* präsentiert werden
- ◆ Der anschauliche, schematische und äußerst praxisnahe Inhalt soll technische und praktische Informationen zu den für die berufliche Praxis wesentlichen Disziplinen vermitteln
- ◆ Die praktischen Übungen, bei denen der Selbstbewertungsprozess zur Verbesserung des Lernens durchgeführt werden kann
- ◆ Sein besonderer Schwerpunkt liegt auf innovativen Methoden
- ◆ Theoretische Vorträge, Fragen an den Experten, Diskussionsforen zu kontroversen Themen und individuelle Reflexionsarbeit
- ◆ Die Verfügbarkeit des Zugangs zu Inhalten von jedem festen oder tragbaren Gerät mit Internetanschluss



Werden Sie mit diesem privaten Masterstudiengang zum Meister der Programmierung in der Cloud. Wachsen Sie in einem hart umkämpften Sektor“

“

Microsoft Azure, Amazon Web Services und Google Cloud sind die wichtigsten Cloud-Plattformen für Unternehmen. Beherrschen Sie alle ihre Möglichkeiten mit diesem privaten Masterstudiengang. Schreiben Sie sich jetzt ein”

Zu den Dozenten des Programms gehören Fachleute aus der Branche, die ihre Erfahrungen aus ihrer Arbeit in diese Weiterbildung einbringen, sowie anerkannte Spezialisten aus führenden Unternehmen und renommierten Universitäten.

Die multimedialen Inhalte, die mit der neuesten Bildungstechnologie entwickelt wurden, werden den Fachkräften ein situierendes und kontextbezogenes Lernen ermöglichen, d. h. eine simulierte Umgebung, die eine immersive Fortbildung bietet, die auf die Ausführung von realen 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 im Laufe der Fortbildung auftreten. Zu diesem Zweck werden sie von einem innovativen interaktiven Videosystem unterstützt, das von renommierten Experten entwickelt wurde.

Vertiefen Sie Ihr Wissen und werden Sie zum Spezialisten für Cloud-Infrastrukturen, indem Sie die neuesten Cloud-Native-Technologien und -Architekturen beherrschen.

Studieren Sie in Ihrem eigenen Tempo, ohne feste Termine und von überall aus mit der Online-Methode, die TECH für alle ihre Abschlüsse anbietet.



02 Ziele

Das Programm dieses privaten Masterstudiengangs versetzt die Studenten in die Lage, in ihrem eigenen *Start-up* oder in großen Unternehmen ein Projekt in *Cloud*-Umgebungen zu starten. Zu diesem Zweck erwerben sie in diesem Studiengang das nötige Fachwissen, um die geeignete *Cloud* zu bestimmen, lernen die verschiedenen Ansätze zur Cloud-Implementierung und deren Zusammenhänge zu analysieren und beherrschen die wichtigsten Tools und Anbieter in diesem aufstrebenden Technologiebereich. Interaktive Inhalte und Fallstudien ermöglichen es ihnen, ihr Wissen in einem der Technologiebereiche mit dem größten Potenzial zu vertiefen.



“

Fördern Sie Ihre Karriere. Jeden Tag benötigen Unternehmen mehr Front-End- oder Back-End-Ingenieure. Spezialisieren Sie sich auf einen Bereich, in dem es viele Beschäftigungsmöglichkeiten gibt”

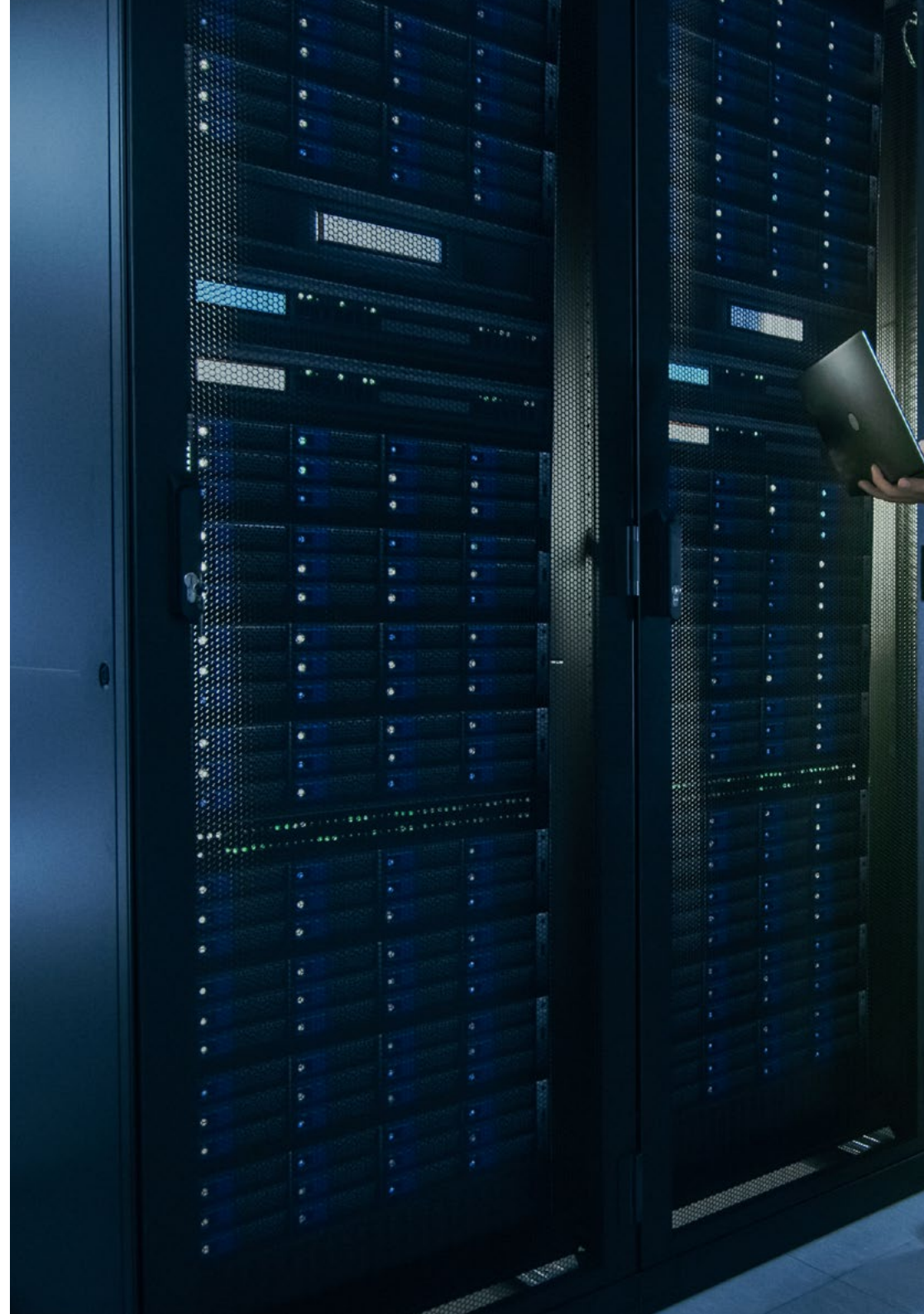


Allgemeine Ziele

- ◆ Analysieren der verschiedenen Ansätze zur *Cloud*-Einführung und ihrer Zusammenhänge
- ◆ Erwerben von Fachwissen zur Bestimmung der geeigneten *Cloud*
- ◆ Entwickeln einer virtuellen Maschine in Azure
- ◆ Ermitteln der Gefahrenquellen bei der Anwendungsentwicklung und der anzuwendenden *Best Practices*
- ◆ Bewerten der Unterschiede in den spezifischen Implementierungen der verschiedenen *Public Cloud*-Anbieter
- ◆ Bestimmen der verschiedenen Technologien, die für Container eingesetzt werden
- ◆ Identifizieren der wichtigsten Aspekte einer Strategie zur Einführung von *Cloud Native*
- ◆ Vermitteln von Grundlagen und Bewerten der im Bereich *Big Data* am häufigsten verwendeten Programmiersprachen, die für die Datenanalyse und -verarbeitung erforderlich sind



Die Multimedia-Ressourcen und das Relearning-System werden Ihnen helfen, Ihr Wissen zu erweitern. Bringen Sie Ihre berufliche Karriere voran, indem Sie ein Experte für Cloud Programming werden“





Spezifische Ziele

Modul 1. Programmierung in der Cloud. Dienste in Azure, AWS und Google Cloud

- ◆ Erwerben von Fachwissen über die *Cloud* und die Unterschiede zu traditionellen *On-Premise*-Lösungen
- ◆ Aneignen eines grundlegenden Fachvokabulars in der *Cloud*. Beherrschen der von den verschiedenen Anbietern verwendeten Begriffe
- ◆ Feststellen der wichtigsten Komponenten der *Cloud* und ihrer Verwendung
- ◆ Bestimmen der Anbieter auf dem *Cloud*-Markt, ihrer Stärken und Schwächen sowie ihrer Beiträge

Modul 2. Programmierung von Architekturen für Cloud Computing

- ◆ Entwickeln von Fachwissen über die Grundlagen der Architektur
- ◆ Spezialisieren der Studenten auf das Wissen über *Cloud*-Infrastrukturen
- ◆ Abwägen der Vor- und Nachteile einer Bereitstellung *On Premise* oder in der *Cloud*
- ◆ Ermitteln der Infrastrukturanforderungen
- ◆ Identifizieren von Bereitstellungsoptionen
- ◆ Fortbilden der Studenten in der Implementierung einer *Cloud*-Infrastruktur
- ◆ Entwerfen und Definieren des Betriebs und der Wartung einer *Cloud*-Architektur

Modul 3. Storage in Cloud Azure

- ◆ Untersuchen einer virtuellen Maschine in Azure
- ◆ Festlegen der verschiedenen Speichertypen
- ◆ Bewerten von Backup-Funktionen
- ◆ Verwalten von Azure-Ressourcen
- ◆ Analysieren der verschiedenen Arten von Diensten
- ◆ Untersuchen der verschiedenen Arten von Sicherheit
- ◆ Generieren von virtuellen Netzwerken
- ◆ Erkennen der verschiedenen Netzwerkverbindungen

Modul 4. Cloud-Umgebungen. Sicherheit

- ◆ Identifizieren der Risiken bei der Bereitstellung einer öffentlichen *Cloud*-Infrastruktur
- ◆ Analysieren von Sicherheitsrisiken bei der Entwicklung von Anwendungen
- ◆ Bestimmen der Sicherheitsanforderungen
- ◆ Entwickeln eines Sicherheitsplans für eine *Cloud*-Bereitstellung
- ◆ Erstellen von Richtlinien für ein *Logging*- und Überwachungssystem
- ◆ Vorschlagen von Maßnahmen zur Reaktion auf Vorfälle

Modul 5. Container-Orchestrierung: Kubernetes und Docker

- ◆ Entwickeln der Grundlagen der Container-Architektur und -Technologie
- ◆ Festlegen der verschiedenen Container-Technologien
- ◆ Bestimmen der Anforderungen an die Infrastruktur
- ◆ Prüfen von Einsatzoptionen

Modul 6. Programmierung Cloud-nativer Anwendungen

- ◆ Einführen der Technologien für Entwicklung und kontinuierliche Integration
- ◆ Demonstrieren, wie Kubernetes als Service-Orchestrator funktioniert
- ◆ Analysieren der Beobachtbarkeit und der Sicherheitstools in *Cloud Native*
- ◆ Evaluieren von Bereitstellungsplattformen
- ◆ Vermitteln der Grundlagen von Datenverwaltungsstrategien in *Cloud Native*-Umgebungen
- ◆ Identifizieren gängiger Techniken in *Cloud Native*-Entwicklungen

Modul 7. Programmierung in der Cloud. Data Governance

- ◆ Erarbeiten von Fachwissen über Datenmanagement, Strategien und Verarbeitungstechniken
- ◆ Entwickeln von Data-Governance-Strategien, die auf Menschen, Prozesse und Tools ausgerichtet sind
- ◆ Durchführen von *Data Governance* von der Aufnahme bis zur Aufbereitung und Nutzung
- ◆ Bestimmen von Techniken zur Steuerung der Datenübertragung
- ◆ Einrichten von Datenschutz für Authentifizierung, Sicherheit, Backup und Überwachung

Modul 8. Programmierung in der Cloud in Echtzeit. Streaming

- ◆ Analysieren des Prozesses der Erfassung, Strukturierung, Verarbeitung, Analyse und Interpretation von *Streaming*-Daten
- ◆ Entwickeln der Prinzipien der *Streaming*-Verarbeitung, des aktuellen Kontextes und der aktuellen Anwendungsfälle im nationalen Kontext
- ◆ Entwickeln wichtiger Grundlagen der Statistik, des *Machine Learning*, des *Data Mining* und der prädiktiven Modellierung für das Verständnis der Datenanalyse und -verarbeitung
- ◆ Analysieren der wichtigsten *Big Data*-Programmiersprachen
- ◆ Untersuchen der Grundlagen von Apache Spark Streaming, Kafka Stream und Flink Stream



Modul 9. Cloud-Integration mit Webdiensten. Technologien und Protokolle

- ◆ Bewerten des Fortschritts von Web-Technologien und -Architekturen, um die Komplexität des Systems zu bestimmen und auf dieser Grundlage eine Softwarelösung vorzuschlagen
- ◆ Entwickeln von verteilten Projekten im *Cloud Computing* unter Verwendung von Webdiensten und unterschiedlichen Funktions- und Sicherheitsanforderungen
- ◆ Analysieren verschiedener Technologien zur Implementierung von Webservices, um diejenige zu identifizieren, die das Problemszenario am besten unterstützt
- ◆ Bewerten der Korrektheit der Implementierung eines serverseitigen Webdienstes, indem Sie Anfragen von verschiedenen Arten von Webclients starten

Modul 10. Programmierung in der Cloud. Projektmanagement und Produktüberprüfung

- ◆ Verstehen der Szenarien und Anwendungen im Lebenszyklusmanagement
- ◆ Verwalten von Projekten als Prozess und Bestimmen des Organisationsmodells
- ◆ Bestimmen der Risiken und Kosten durch Anwendung agiler Methoden in der Konzeptionsphase oder während der Projektdurchführung
- ◆ Führen und Verwalten von Projekten mit agilen Methoden und die Qualität von *Cloud*-Projekten durch die Anwendung verschiedener Methoden

03

Kompetenzen

Dieser private Masterstudiengang ist so konzipiert, dass die IT-Fachkräfte, an die er sich richtet, in der Lage sind, die *Cloud*-Programmierung in Unternehmen durchzuführen, die ihre gesamten Ressourcen auf agile Weise verwalten möchten. Am Ende des Studiums werden sie in der Lage sein, die verschiedenen bestehenden Anbieter zu verwalten und den Einsatz von Virtualisierungstechnologien, *Cloud Computing* und Containern zu optimieren und zu differenzieren. Die Dozenten werden Sie während des gesamten Studiums zu Spitzenleistungen führen.



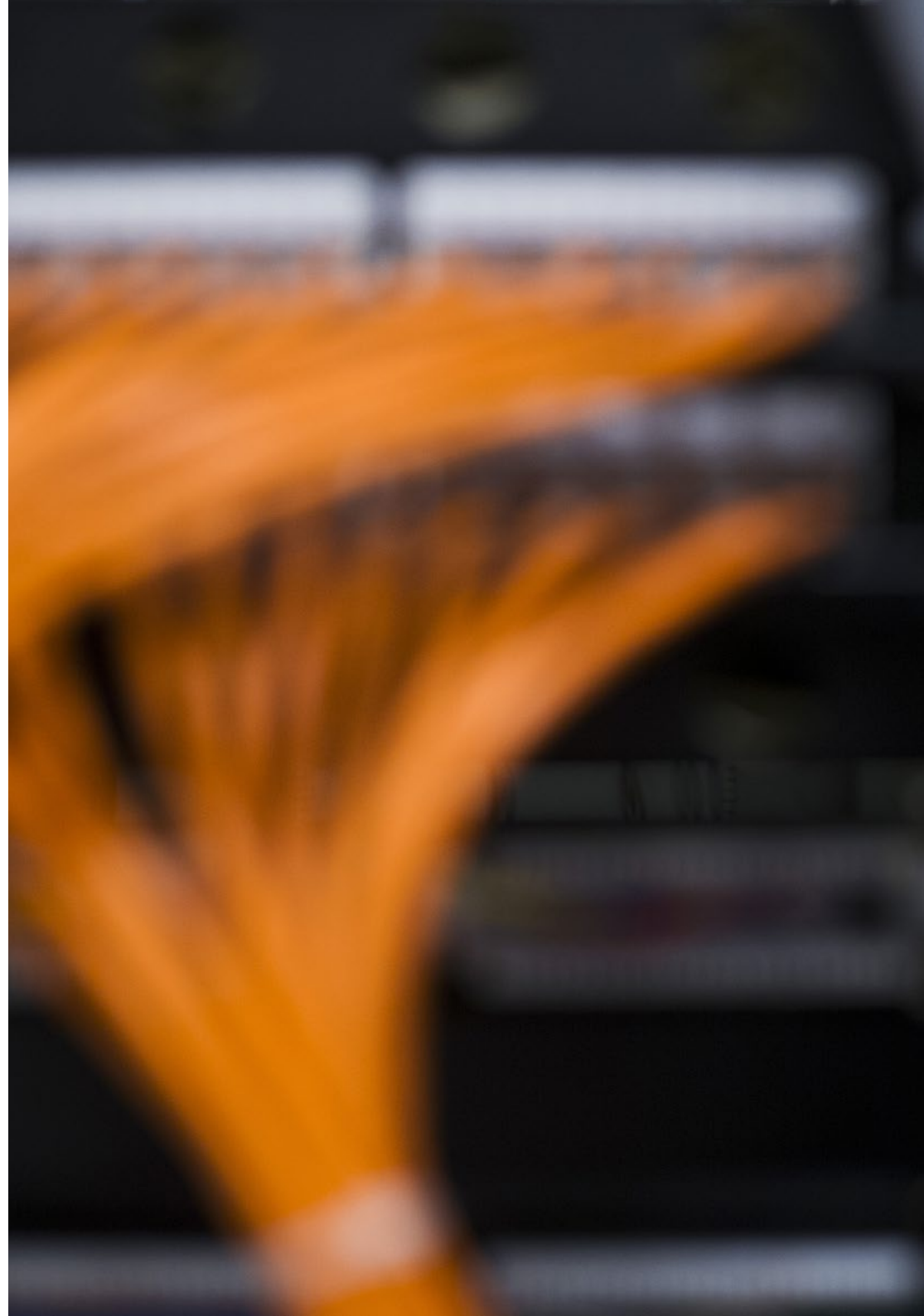
“

Sie erwerben die Grundlagen der DevOp-Methodik und wenden diese auf die Entwicklung digitaler Produkte an”



Allgemeine Kompetenzen

- ◆ Analysieren des Transformationsprozesses von Unternehmen, die die DevOps-Methodik eingeführt haben
- ◆ Erlernen der Grundlagen der DevOps-Methodik, um diese bei der Steuerung des Implementierungs- und Entwicklungsprozesses digitaler Produkte anzuwenden
- ◆ Beherrschen der verschiedenen existierenden Business-Analyse-Techniken zur korrekten Entwicklung der Planungsphase
- ◆ Festlegen der bestehenden Verifikations- und Validierungstechniken, um die Qualität des entwickelten Produkts sicherzustellen
- ◆ Identifizieren der Unterschiede zwischen Virtualisierung, *Cloud Computing* und Container-Technologien, um jede dieser Technologien optimal nutzen zu können
- ◆ Untersuchen der Grundlagen, auf denen *Cloud Services* basieren, um diese Werkzeuge im Unternehmen richtig einsetzen zu können
- ◆ Kennen der Anbieter und Merkmale von *Cloud Computing*-Diensten, um diejenigen auszuwählen, die den Bedürfnissen des Unternehmens am besten entsprechen





Spezifische Kompetenzen

- ◆ Identifizieren der wichtigsten Datenverarbeitungstechniken
- ◆ Erweitern der Kenntnisse über die verschiedenen Tools und deren Einsatz im Bereich des Projektmanagements
- ◆ Generieren von Fachwissen über Servicequalität und wie man eine angemessene Qualität mit seinem Produkt erreicht
- ◆ Begründen des konkreten Einsatzes einer Servicearchitektur, um eine Lösung für ein Problem in einem konkreten Rahmen zu bieten
- ◆ Identifizieren von Schwachstellen und Bedrohungen eines Systems, um eine technologische Lösung vorzuschlagen, die die Sicherheit eines Systems unterstützt
- ◆ Untersuchen der verschiedenen von *Cloud*-Anbietern bereitgestellten Dienste und Begründen ihres Einsatzes in einem konkreten Projekt
- ◆ Untersuchen der Verwendung von Containern und der Entwicklung mit Microservices
- ◆ Identifizieren der *Cloud*-Dienste, die für die Umsetzung eines Sicherheitsplans eingesetzt werden sollen, und der für Präventionsmechanismen erforderlichen Vorgänge

04

Kursleitung

Im Rahmen ihrer Maxime, eine Elitefortbildung für alle anzubieten, wählt die TECH ihr Dozententeam sorgfältig aus, um den Studenten solide Kenntnisse im Bereich der *Cloud*-Programmierung zu vermitteln. So verfügt dieser private Masterstudiengang über ein hoch qualifiziertes Team mit umfassender Erfahrung in diesem Bereich. Dank ihrer Kenntnisse und ihrer Nähe zum Berufsfeld bringen die Inhalte dieses Studiengangs die IT-Profis mit den neuesten Entwicklungen dieser Technologie und den aktuellen Bedürfnissen der Unternehmen in diesem Bereich in Kontakt.



“

Ein leistungsstarkes Dozententeam begleitet Sie während der zwölf Monate dieses privaten Masterstudiengangs. Anklicken und einschreiben"

Leitung



Hr. Bressel Gutiérrez-Ambrossi, Guillermo

- Spezialist für Computersysteme und Netzwerkadministration
- Speicher- und SAN-Netzwerkadministrator bei Experis IT (BBVA)
- Netzwerkadministrator bei der IE Business School
- Hochschulabschluss in Computersystemen und Netzwerkadministration an der ASIR
- Ethical Hacking-Kurs bei OpenWebinar
- Powershell-Kurs bei OpenWebinar

Professoren

Hr. Gómez Rodríguez, Antonio

- ◆ Leitender Ingenieur für Cloud-Lösungen bei Oracle
- ◆ Mitorganisator des Malaga Developer Meetup
- ◆ Beratungsspezialist für die Sopra Group und Everis
- ◆ Teamleiter bei System Dynamics
- ◆ Software-Entwickler bei SGO Software
- ◆ Masterstudiengang in E-Business an der La Salle-Wirtschaftsschule
- ◆ Aufbaustudiengang in Informationstechnologien und -systemen, Katalanisches Institut für Technologie
- ◆ Hochschulabschluss in Telekommunikationstechnik an der Polytechnischen Universität von Katalonien

Hr. Bernal de la Varga, Yeray

- ◆ Architekt für *Big Data*-Lösungen bei der Orange Bank
- ◆ *Big-Data*-Architekt bei Bankia
- ◆ Ingenieur für *Big Data* bei Hewlett-Packard
- ◆ Außerordentlicher Professor im Masterstudiengang *Big Data* an der Universität von Deusto
- ◆ Hochschulabschluss in Informatik an der Polytechnischen Universität von Madrid
- ◆ Experte in *Big Data* von der U-tad

Fr. Rodríguez Camacho, Cristina

- ◆ Apis-Beraterin und Microservices-Entwicklerin bei Inetum
- ◆ Hochschulabschluss in Gesundheitsingenieurwesen mit Spezialisierung auf Biomedizintechnik von der Universität von Málaga
- ◆ Masterstudiengang in *Blockchain* und *Big Data* von der Universität Complutense von Madrid
- ◆ Experte in Devops & Cloud an der UNIR

Hr. Torres Palomino, Sergio

- ◆ IT-Ingenieur mit Spezialisierung auf Blockchain
- ◆ *Blockchain Lead* bei Telefónica
- ◆ *Blockchain*-Architekt bei Signeblock
- ◆ *Blockchain*-Entwickler bei Blocknitive
- ◆ Autor und Kommunikator bei O'Really Media Books
- ◆ Dozent für Aufbaustudiengänge und *Blockchain*-bezogene Kurse
- ◆ Hochschulabschluss in Computertechnik von der Universität San Pablo CEU
- ◆ Masterstudiengang in *Big Data*-Architektur
- ◆ Masterstudiengang in *Big Data* und *Business Analytics*

Hr. Rodríguez García, Darío

- ◆ Softwarearchitekt bei NEA F3 Master
- ◆ Full-Stack-Entwickler bei NEA F3 Master
- ◆ Hochschulabschluss in Computer Software Engineering an der Universität von Oviedo
- ◆ Masterstudiengang in Web Engineering an der Universität von Oviedo
- ◆ Professor für Web-Engineering-Programme
- ◆ Dozent für Kurse auf der E-Learning-Plattform Udemy

Dr. Moguel Márquez, Miguel

- ◆ Computer-Ingenieur und Technologie-Berater
- ◆ Berater im Bereich Web Engineering, Design und Entwicklung von Webanwendungen, Software-Architekturen und neue technologische Trends
- ◆ Promotion in Computertechnologien an der Universität von Extremadura
- ◆ Masterstudiengang in Computertechnik an der Universität von Extremadura
- ◆ Hochschulabschluss in Informatik an der Universität von Extremadura

Dr. García Sanz-Calcedo, Justo

- ◆ Fachingenieur für Gesundheit
- ◆ Direktor für Technik und Wartung im Gesundheitsdienst von Extremadura
- ◆ Promotion in Wirtschaftsingenieurwesen an der Universität von Extremadura
- ◆ Hochschulabschluss in Wirtschaftsingenieurwesen an der Universität von Extremadura
- ◆ Experte für Teammanagement-Fähigkeiten und Ausbilder von Ausbildern
- ◆ Programm für leitendes Management in Gesundheitseinrichtungen an der IESE Business School





Dr. Sánchez-Barroso Moreno, Gonzalo

- ◆ Industrie- und Maschinenbauingenieur
- ◆ Berater für industrielle Forschung und experimentelle Entwicklungsprojekte
- ◆ Promotion in Wirtschaftsingenieurwesen an der Universität von Extremadura
- ◆ Hochschulabschluss in Maschinenbau an der Universität von Extremadura
- ◆ Masterstudiengang in Wirtschaftsingenieurwesen an der Universität von Extremadura
- ◆ Spezialisierung auf das Management von Innovationsprojekten
- ◆ Certified Project Management Associate (Level D) von der International Project Management Association (IPMA)

Dr. González Domínguez, Jaime

- ◆ Berater für industrielle Forschung und experimentelle Entwicklungsprojekte
- ◆ Promotion in Modellierung und Experimentierung in Wissenschaft und Technik
- ◆ Wirtschaftsingenieur und Maschinenbauingenieur von der Universität von Extremadura
- ◆ Spezialisierung auf das Management von Innovationsprojekten
- ◆ Certified Project Management Associate (Level D) von der International Project Management Association (IPMA)

05

Struktur und Inhalt

Der Lehrplan wurde auf der Grundlage der umfassenden Anforderungen des Dozententeams entwickelt, aus dem sich der Studiengang zusammensetzt. Das Ergebnis ist ein aus zehn Modulen bestehender Lehrplan, der einen breiten und detaillierten Überblick über *Cloud*-Umgebungen, die verschiedenen verfügbaren Tools und ihre Möglichkeiten in einem aufstrebenden Bereich bietet. IT-Fachkräfte, die diesen Kurs absolvieren, werden in der Lage sein, *Cloud-Native*-Anwendungen zu programmieren, ein sicheres Netzwerk zu entwerfen und zu implementieren oder Programmierung in der Cloud in Echtzeit durchzuführen. All dies wird durch umfangreiche Multimedia-Inhalte mit detaillierten Videos, weiterführender Lektüre und praktischen Beispielen aus der Praxis unterstützt, die diesen Kurs ergänzen.



DEV



OPS

“

TECH bietet Ihnen einen Lehrplan mit hochwertigen Inhalten und einem aktuellen und innovativen Ansatz für Cloud-Umgebungen”

Modul 1. Programmierung in der Cloud. Dienste in Azure, AWS und Google Cloud

- 1.1. *Cloud*. *Cloud*-Dienste und -Technologie
 - 1.1.1. *Cloud*-Dienste und -Technologie
 - 1.1.2. *Cloud*-Terminologie
 - 1.1.3. Referenz-*Cloud*-Anbieter
- 1.2. *Cloud Computing*
 - 1.2.1. *Cloud Computing*
 - 1.2.2. *Cloud Computing*-Ökosystem
 - 1.2.3. *Cloud Computing*-Typologie
- 1.3. *Cloud*-Dienstmodelle
 - 1.3.1. IaaS. Infrastruktur als Dienstleistung
 - 1.3.2. SaaS. Software als Dienstleistung
 - 1.3.3. PaaS. Plattform als Dienstleistung
- 1.4. *Cloud Computing*-Technologien
 - 1.4.1. Virtualisierungssystem
 - 1.4.2. Service-Oriented Architecture (SOA)
 - 1.4.3. GRID-Computing
- 1.5. Architektur des *Cloud Computing*
 - 1.5.1. Architektur des *Cloud Computing*
 - 1.5.2. *Cloud Computing*-Netzwerk-Typologien
 - 1.5.3. Sicherheit im *Cloud Computing*
- 1.6. *Public Cloud*
 - 1.6.1. *Public Cloud*
 - 1.6.2. Architektur und Kosten der *Public Cloud*
 - 1.6.3. *Public Cloud*. Typologie
- 1.7. *Private Cloud*
 - 1.7.1. *Private Cloud*
 - 1.7.2. Architektur und Kosten
 - 1.7.3. *Private Cloud*. Typologie
- 1.8. *Hybrid Cloud*
 - 1.8.1. *Hybrid Cloud*
 - 1.8.2. Architektur und Kosten
 - 1.8.3. *Hybrid Cloud*. Typologie

- 1.9. *Cloud*-Anbieter
 - 1.9.1. Amazon Web Services
 - 1.9.2. Azure
 - 1.9.3. Google
- 1.10. *Cloud*-Sicherheit
 - 1.10.1. Sicherheit der Infrastruktur
 - 1.10.2. Betriebssystem- und Netzwerksicherheit
 - 1.10.3. Risikominderung in der *Cloud*

Modul 2. Programmierung von Architekturen für *Cloud Computing*

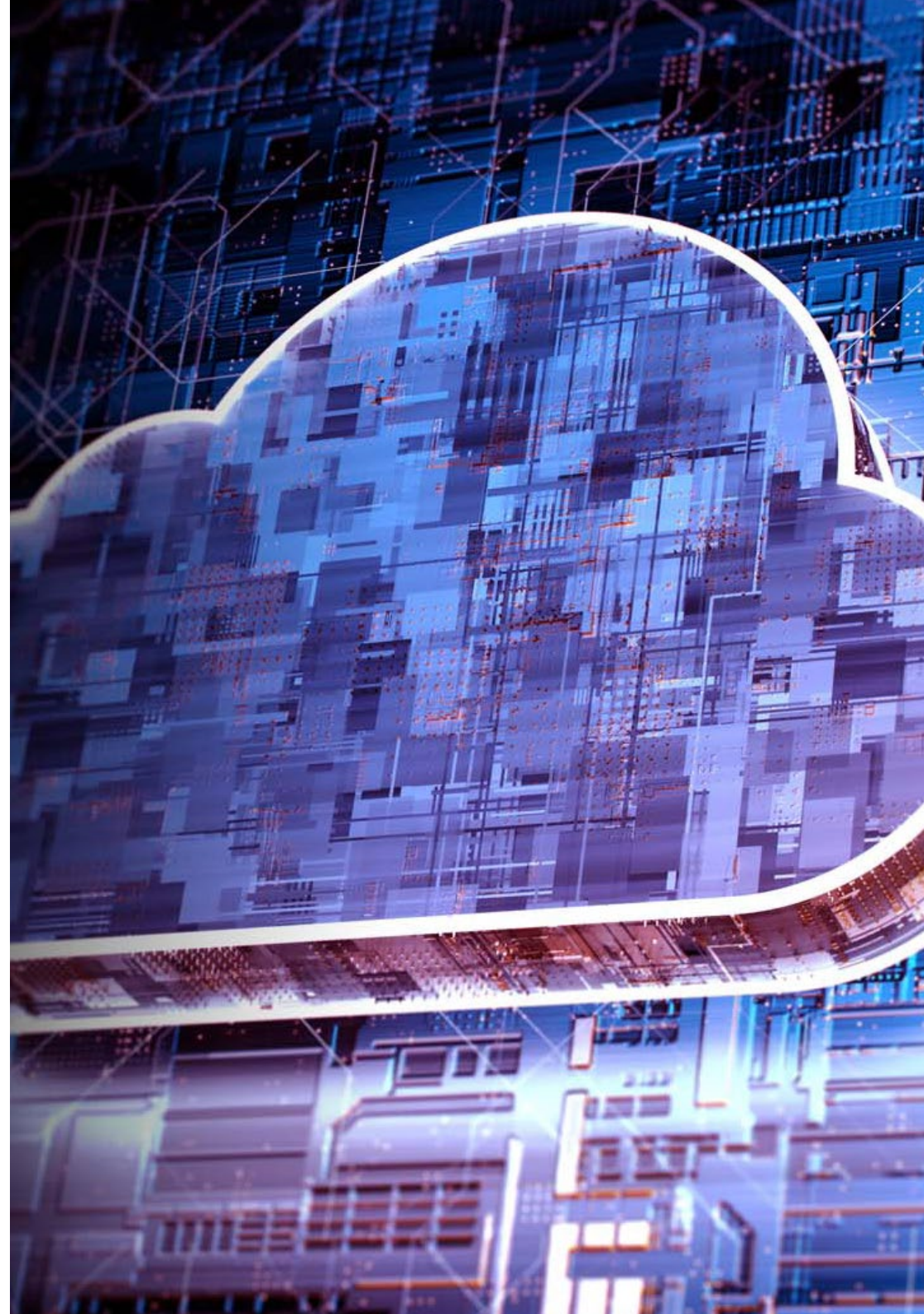
- 2.1. *Cloud*-Architektur für ein Universitätsnetzwerk. Auswahl des *Cloud*-Anbieters. Praktisches Beispiel
 - 2.1.1. *Cloud*-Architekturansatz für ein Universitätsnetzwerk nach *Cloud*-Anbieter
 - 2.1.2. Komponenten der *Cloud*-Architektur
 - 2.1.3. Analyse von *Cloud*-Lösungen gemäß der vorgeschlagenen Architektur
- 2.2. Wirtschaftliche Schätzung des Projekts zur Schaffung eines Universitätsnetzwerks. Finanzierung
 - 2.2.1. Auswahl des *Cloud*-Anbieters
 - 2.2.2. Wirtschaftliche Schätzung auf der Grundlage der Komponenten
 - 2.2.3. Finanzierung des Projekts
- 2.3. Schätzung des Personalbedarfs für das Projekt. Zusammensetzung eines Softwareteams
 - 2.3.1. Zusammensetzung des Softwareentwicklungsteams
 - 2.3.2. Rollen in einem Entwicklungsteam. Typologie
 - 2.3.3. Bewertung der wirtschaftlichen Einschätzung des Projekts
- 2.4. Umsetzungszeitplan und Projektdokumentation
 - 2.4.1. Agiler Projektzeitplan
 - 2.4.2. Dokumentation der Projektdurchführbarkeit
 - 2.4.3. Die für die Durchführung des Projekts zu erstellende Dokumentation
- 2.5. Rechtliche Implikationen eines Projekts
 - 2.5.1. Rechtliche Implikationen eines Projekts
 - 2.5.2. Datenschutzpolitik
 - 2.5.2.1. DSGVO. Datenschutz-Grundverordnung
 - 2.5.3. Verantwortung des integrierenden Unternehmens

- 2.6. Entwurf und Erstellung eines *Cloud-Blockchain*-Netzwerks für die vorgeschlagene Architektur
 - 2.6.1. *Blockchain - Hyperledger Fabric*
 - 2.6.2. *Hyperledger Fabric Basics*
 - 2.6.3. Aufbau eines internationalen universitären *Hyperledger Fabric*-Netzwerks
 - 2.7. Vorgeschlagener Ansatz zur Erweiterung der Architektur
 - 2.7.1. Erstellung der vorgeschlagenen Architektur mit *Blockchain*
 - 2.7.2. Erweitern der vorgeschlagenen Architektur
 - 2.7.3. Konfiguration einer hochverfügbaren Architektur
 - 2.8. Verwaltung der vorgeschlagenen *Cloud*-Architektur
 - 2.8.1. Hinzufügen eines neuen Teilnehmers zur ursprünglich vorgeschlagenen Architektur
 - 2.8.2. Verwaltung der *Cloud*-Architektur
 - 2.8.3. Verwaltung der Projektlogik - *Smart Contracts*
 - 2.9. Verwaltung und Management der spezifischen Komponenten der vorgeschlagenen *Cloud*-Architektur
 - 2.9.1. Verwaltung von Netzwerkzertifikaten
 - 2.9.2. Verwaltung der Sicherheit der verschiedenen Komponenten: CouchDB
 - 2.9.3. Verwaltung von *Blockchain*-Netzwerkknoten
 - 2.10. Änderung einer anfänglichen Basisinstallation bei der Erstellung des *Blockchain*-Netzwerks
 - 2.10.1. Hinzufügen eines Knotens zum *Blockchain*-Netzwerk
 - 2.10.2. Hinzufügen von zusätzlichen Datenpersistenzen
 - 2.10.3. Verwaltung von *Smart Contracts*
 - 2.10.4. Hinzufügen einer neuen Universität zu einem bestehenden Netzwerk
-
- Modul 3. Storage in Cloud Azure**
- 3.1. MV-Installation in Azure
 - 3.1.1. Befehle zur Erstellung
 - 3.1.2. Befehle zur Ansicht
 - 3.1.3. Befehle zum Ändern
 - 3.2. *Blobs* in Azure
 - 3.2.1. *Blob*-Typen
 - 3.2.2. Container
 - 3.2.3. *Azcopy*
 - 3.2.4. Umkehrbare *Blob*-Unterdrückung
 - 3.3. Verwaltete Festplatten und Speicher in Azure
 - 3.3.1. Verwaltete Festplatte
 - 3.3.2. Sicherheit
 - 3.3.3. Kalter Speicher
 - 3.3.4. Replikation
 - 3.3.4.1. Lokale Redundanz
 - 3.3.4.2. Redundanz in einem Bereich
 - 3.3.4.3. "Georedundante"
 - 3.4. Tabellen, Warteschlangen, Dateien in Azure
 - 3.4.1. Tabellen
 - 3.4.2. Warteschlangen
 - 3.4.3. Dateien
 - 3.5. Verschlüsselung und Sicherheit in Azure
 - 3.5.1. *Storage Service Encryption* (SSE)
 - 3.5.2. Zugriffsschlüssel
 - 3.5.2.1. Gemeinsame Zugriffssignatur
 - 3.5.2.2. Zugriffsrichtlinien auf Containerebene
 - 3.5.2.3. Zugriffssignatur auf *Blob*-Ebene
 - 3.5.3. Azure AD-Authentifizierung
 - 3.6. Virtuelles Netzwerk in Azure
 - 3.6.1. Subnetting und Pairing
 - 3.6.2. *Vnet to Vnet*
 - 3.6.3. Private Verbindung
 - 3.6.4. Hohe Verfügbarkeit
 - 3.7. Arten von Verbindungen in Azure
 - 3.7.1. *Azure Application Gateway*
 - 3.7.2. Standort-zu-Standort-VPN
 - 3.7.3. Punkt-zu-Standort-VPN
 - 3.7.4. ExpressRoute
 - 3.8. Ressourcen in Azure
 - 3.8.1. Ressourcen sperren
 - 3.8.2. Ressourcen-Bewegung
 - 3.8.3. Entsorgung von Ressourcen

- 3.9. *Backup in Azure*
 - 3.9.1. *Recovery Services*
 - 3.9.2. *Agent Azure Backup*
 - 3.9.3. *Azure Backup Server*
- 3.10. *Entwicklung von Lösungen*
 - 3.10.1. *Komprimierung, Deduplizierung, Replikation*
 - 3.10.2. *Recovery Services*
 - 3.10.3. *Disaster Recovery Plan*

Modul 4. *Cloud-Umgebungen. Sicherheit*

- 4.1. *Cloud-Umgebungen. Sicherheit*
 - 4.1.1. *Cloud-Umgebungen, Sicherheit*
 - 4.1.1.1. *Cloud-Sicherheit*
 - 4.1.1.2. *Sicherheitsposition*
- 4.2. *Gemeinsames Sicherheitsverwaltungsmodell in der Cloud*
 - 4.2.1. *Vom Anbieter verwaltete Sicherheitselemente*
 - 4.2.2. *Vom Kunden verwaltete Elemente*
 - 4.2.3. *Sicherheitsstrategien*
- 4.3. *Cloud-Präventionsmechanismen*
 - 4.3.1. *Authentifizierungsmanagementsysteme*
 - 4.3.2. *Authentifizierungsmanagementsystem: Zugangspolitik*
 - 4.3.3. *Systeme zur Schlüsselverwaltung*
- 4.4. *Datensicherheit in der Cloud-Infrastruktur*
 - 4.4.1. *Absicherung von Speichersystemen:*
 - 4.4.1.1. *Block*
 - 4.4.1.2. *Object Storage*
 - 4.4.1.3. *File Systems*
 - 4.4.2. *Sicherung von Datenbanksystemen*
 - 4.4.3. *Sichern von Daten bei der Übermittlung*
- 4.5. *Schutz der Cloud-Infrastruktur*
 - 4.5.1. *Entwurf und Implementierung eines sicheren Netzwerks*
 - 4.5.2. *Sicherheit von Computerressourcen*
 - 4.5.3. *Tools und Ressourcen zum Schutz der Infrastruktur*



- 4.6. Anwendungsrisiken und Schwachstellen
 - 4.6.1. Risiken bei der Anwendungsentwicklung
 - 4.6.2. Kritische Sicherheits-Risiken
 - 4.6.3. Schwachstellen in der Softwareentwicklung
- 4.7. Verteidigung von Anwendungen gegen Angriffe
 - 4.7.1. Design in der Anwendungsentwicklung
 - 4.7.2. Absicherung durch Verifizierung und Testen
 - 4.7.3. Sichere Programmierpraxis
- 4.8. Sicherheit in DevOps-Umgebungen
 - 4.8.1. Sicherheit in virtualisierten und containerisierten Umgebungen
 - 4.8.2. Sicherheit in Entwicklung und Betrieb (DevSecOps)
 - 4.8.3. Bewährte Sicherheitsverfahren in Produktionsumgebungen mit Containern
- 4.9. Sicherheit in öffentlichen *Clouds*
 - 4.9.1. AWS
 - 4.9.2. Azure
 - 4.9.3. Oracle *Cloud*
- 4.10. Sicherheitsvorschriften, *Governance* und *Compliance*
 - 4.10.1. *Compliance* im Bereich Sicherheit
 - 4.10.2. Risikomanagement
 - 4.10.3. Prozesse in Organisationen

Modul 5. Container-Orchestrierung: Kubernetes und Docker

- 5.1. Grundlage von Anwendungsarchitekturen
 - 5.1.1. Aktuelle Anwendungsmodelle
 - 5.1.2. Ausführungsplattformen für Anwendungen
 - 5.1.3. Container-Technologien
- 5.2. Docker-Architektur
 - 5.2.1. Docker-Architektur
 - 5.2.2. Installation der Docker-Architektur
 - 5.2.3. Befehle. Lokales Projekt
- 5.3. Docker-Architektur. Speicherverwaltung
 - 5.3.1. Bildverwaltung und Registrierung
 - 5.3.2. Vernetzung in Docker
 - 5.3.3. Speicherverwaltung
- 5.4. Erweiterte Docker-Architektur
 - 5.4.1. Docker *Compose*
 - 5.4.2. Docker in der Organisation
 - 5.4.3. Beispiel für die Einführung von Docker
- 5.5. Kubernetes-Architektur
 - 5.5.1. Kubernetes-Architektur
 - 5.5.2. Kubernetes-Bereitstellungselemente
 - 5.5.3. Distributionen und verwaltete Lösungen
 - 5.5.4. Installation und Umgebung
- 5.6. Kubernetes-Architekturen: Entwickeln mit Kubernetes
 - 5.6.1. Tools für die K8s-Entwicklung
 - 5.6.2. Imperativer vs. deklarativer Modus
 - 5.6.3. Bereitstellung und Exposition von Anwendungen
- 5.7. Kubernetes in Unternehmensumgebungen
 - 5.7.1. Persistenz von Daten
 - 5.7.2. Hochverfügbarkeit, Skalierung und Networking
 - 5.7.3. Sicherheit in Kubernetes
 - 5.7.4. Kubernetes Verwaltung und Überwachung
- 5.8. K8s-Distributionen
 - 5.8.1. Vergleich der Bereitstellungsumgebungen
 - 5.8.2. Bereitstellung auf GKE, AKS, EKS oder OKE
 - 5.8.3. *On-Premise*-Bereitstellung
- 5.9. *Rancher* und *OpenShift*
 - 5.9.1. *Rancher*
 - 5.9.2. *OpenShift*
 - 5.9.3. *OpenShift*: Anwendungskonfiguration und -bereitstellung
- 5.10. Kubernetes und Container-Architekturen. Aktualisierungen
 - 5.10.1. *Open Application Model*
 - 5.10.2. Tools für die Verwaltung der Bereitstellung in Kubernetes-Umgebungen
 - 5.10.3. Referenzen zu anderen Projekten und Trends

Modul 6. Programmierung *Cloud*-nativer Anwendungen

- 6.1. *Cloud Native*-Technologien
 - 6.1.1. *Cloud Native*-Technologien
 - 6.1.2. *Cloud Native Computing Foundation*
 - 6.1.3. *Cloud Native*-Entwicklungswerkzeuge
- 6.2. *Cloud Native*-Anwendungsarchitektur
 - 6.2.1. Entwurf einer *Cloud Native*-Anwendung
 - 6.2.2. Komponenten der *Cloud Native*-Architektur
 - 6.2.3. Modernisierung von *Legacy*-Anwendungen
- 6.3. *Containerization*
 - 6.3.1. Container-orientierte Entwicklung
 - 6.3.2. Entwicklung mit *Microservices*
 - 6.3.3. Tools für die Teamarbeit
- 6.4. DevOps und kontinuierliche Integration und Bereitstellung
 - 6.4.1. Kontinuierliche Integration und Bereitstellung: CI/CD
 - 6.4.2. Ökosystem von Tools für CI/CD
 - 6.4.3. Erstellen einer CI/CD-Umgebung
- 6.5. Beobachtbarkeit und Analyse der Plattform
 - 6.5.1. Beobachtbarkeit von *Cloud Native*-Anwendungen
 - 6.5.2. Tools für Überwachung, *Logging* und Nachvollziehbarkeit
 - 6.5.3. Implementierung einer Beobachtbarkeits- und Analyseumgebung
- 6.6. Datenverwaltung in *Cloud Native*-Anwendungen
 - 6.6.1. Datenbank in *Cloud Native*
 - 6.6.2. Muster für die Datenverwaltung
 - 6.6.3. Technologien für die Implementierung von Datenverwaltungsmustern
- 6.7. Kommunikation in *Cloud Native*-Anwendungen
 - 6.7.1. Synchrone und asynchrone Kommunikation
 - 6.7.2. Technologien für synchrone Kommunikationsmuster
 - 6.7.3. Technologien für asynchrone Kommunikationsmuster

- 6.8. Ausfallsicherheit, Sicherheit und Leistung in *Cloud Native*-Anwendungen
 - 6.8.1. Ausfallsicherheit von Anwendungen
 - 6.8.2. Sichere Entwicklung in *Cloud Native*-Anwendungen
 - 6.8.3. Anwendungsleistung und Skalierbarkeit
- 6.9. Serverless
 - 6.9.1. Serverless in *Cloud Native*
 - 6.9.2. *Serverless*-Plattformen
 - 6.9.3. Anwendungsfälle für *Serverless*-Entwicklung
- 6.10. Plattformen für die Bereitstellung
 - 6.10.1. *Cloud Native*-Entwicklungsumgebungen
 - 6.10.2. Orchestrierungsplattformen. Vergleich
 - 6.10.3. Infrastruktur-Automatisierung

Modul 7. Programmierung in der *Cloud*. *Data Governance*

- 7.1. Datenverwaltung
 - 7.1.1. Datenverwaltung
 - 7.1.2. Ethik im Umgang mit Daten
- 7.2. *Data Governance*
 - 7.2.1. Einstufung, Zugangskontrolle
 - 7.2.2. Regulierung der Datenverarbeitung
 - 7.2.3. *Data Governance*. Wert
- 7.3. Datenverwaltung. Tools
 - 7.3.1. Abstammung
 - 7.3.2. Metadaten
 - 7.3.3. Datenkatalog. *Business Glossary*
- 7.4. Benutzer und Prozesse in der Datenverwaltung
 - 7.4.1. Benutzer
 - 7.4.1.1. Rollen und Verantwortlichkeiten
 - 7.4.2. Prozesse
 - 7.4.2.1. Datenanreicherung

- 7.5. Lebenszyklus von Unternehmensdaten
 - 7.5.1. Datenerzeugung
 - 7.5.2. Datenverarbeitung
 - 7.5.3. Datenspeicherung
 - 7.5.4. Datennutzung
 - 7.5.5. Vernichtung von Daten
- 7.6. Qualität der Daten
 - 7.6.1. Datenqualität in der Datenverwaltung
 - 7.6.2. Datenqualität in der Analytik
 - 7.6.3. Techniken zur Datenqualität
- 7.7. Datenverwaltung im Transit
 - 7.7.1. Datenverwaltung im Transit
 - 7.7.1.1. Abstammung
 - 7.7.2. Die vierte Dimension
- 7.8. Datenschutz
 - 7.8.1. Ebenen des Zugangs
 - 7.8.2. Klassifizierung
 - 7.8.3. *Compliance*. Vorschriften
- 7.9. Überwachung und Messung der *Data Governance*
 - 7.9.1. Überwachung und Messung der *Data Governance*
 - 7.9.2. Überwachung der Abstammung
 - 7.9.3. Überwachung der Datenqualität
- 7.10. Tools zur Datenverwaltung
 - 7.10.1. Talend
 - 7.10.2. Collibra
 - 7.10.3. Informatik

Modul 8. Programmierung in der Cloud in Echtzeit. *Streaming*

- 8.1. Verarbeitung und Strukturierung von *Streaming*-Informationen
 - 8.1.1. Prozess der Erfassung, Strukturierung, Verarbeitung, Analyse und Interpretation von Daten
 - 8.1.2. Techniken zur Verarbeitung von *Streaming*-Daten
 - 8.1.3. *Streaming*-Verarbeitung
 - 8.1.4. Anwendungsfälle der *Streaming*-Verarbeitung
- 8.2. Statistiken zum Verständnis des gestreamten Datenflusses
 - 8.2.1. Deskriptive Statistik
 - 8.2.2. Berechnung der Wahrscheinlichkeiten
 - 8.2.3. Inferenz
- 8.3. Programmieren mit Python
 - 8.3.1. Typologie, Konditionale, Funktionen und Schleifen
 - 8.3.2. Numpy, Matplotlib, DataFrames, CSV-Dateien und .json-Formate
 - 8.3.3. Sequenzen: Listen, Schleifen, Dateien und Wörterbücher
 - 8.3.4. Veränderlichkeit, Ausnahmen und Funktionen höherer Ordnung
- 8.4. Programmieren mit R
 - 8.4.1. Programmieren mit R
 - 8.4.2. Vektoren und Faktoren
 - 8.4.3. Matrizen und *Arrays*
 - 8.4.4. Listen und *Data Frame*
 - 8.4.5. Funktionen
- 8.5. SQL-Datenbank für *Streaming*-Datenverarbeitung
 - 8.5.1. SQL-Datenbank
 - 8.5.2. Entity-Relationship-Modell
 - 8.5.3. Relationales Modell
 - 8.5.4. SQL
- 8.6. NON-SQL-Datenbank für *Streaming Data Processing*
 - 8.6.1. NO SQL-Datenbank
 - 8.6.2. MongoDB
 - 8.6.3. MongoDB-Architektur
 - 8.6.4. CRUD-Operationen
 - 8.6.5. *Find*, Projektionen, *Indexes-Aggregation* und *Cursors*
 - 8.6.6. Datenmodell

- 8.7. *Data Mining* und prädiktive Modellierung
 - 8.7.1. Multivariate Analyse
 - 8.7.2. Techniken zur Dimensionalitätsreduktion
 - 8.7.3. Cluster-Analyse
 - 8.7.4. Serien
- 8.8. *Maching Learning* für die Verarbeitung von *Streaming*-Daten
 - 8.8.1. *Maching Learning* und erweiterte prädiktive Modellierung
 - 8.8.2. Neuronale Netze
 - 8.8.3. *Deep Learning*
 - 8.8.4. *Bagging* und *Random Forest*
 - 8.8.5. *Gradient Bosting*
 - 8.8.6. SVM
 - 8.8.7. Montage-Methoden
- 8.9. Technologien zur Verarbeitung von *Streaming*-Daten
 - 8.9.1. *Spark Streaming*
 - 8.9.2. *Kafka Streams*
 - 8.9.3. *Flink Streaming*
- 8.10. *Apache Spark Streaming*
 - 8.10.1. *Apache Spark Streaming*
 - 8.10.2. Spark-Komponenten
 - 8.10.3. Spark-Architektur
 - 8.10.4. RDD
 - 8.10.5. SPARK SQL
 - 8.10.6. *Jobs, Stages* und *Task*

Modul 9. *Cloud*-Integration mit Webdiensten. Technologien und Protokolle

- 9.1. Webstandards und -protokolle
 - 9.1.1. Web und Web 2.0
 - 9.1.2. Client-Server-Architektur
 - 9.1.3. Kommunikationsprotokolle und Standards
- 9.2. Webdienste
 - 9.2.1. Webdienste
 - 9.2.2. Kommunikationsschichten und -mechanismen
 - 9.2.3. Dienst-Architekturen



- 9.3. Service-orientierte Architekturen
 - 9.3.1. *Service Oriented Architecture* (SOA)
 - 9.3.2. Entwurf von Webdiensten
 - 9.3.3. SOAP und REST
- 9.4. SOAP. Service Oriented Architecture
 - 9.4.1. Struktur und Nachrichtenübermittlung
 - 9.4.2. *Web Service Description Language* (WSDL)
 - 9.4.3. Implementierung von SOAP-Clients und -Servern
- 9.5. REST-Architektur
 - 9.5.1. REST-Architekturen und RESTful Web Services
 - 9.5.2. HTTP-Verben: Semantik und Zweck
 - 9.5.3. *Swagger*
 - 9.5.4. REST-Client und Server-Implementierungen
- 9.6. Microservices-basierte Architekturen
 - 9.6.1. Monolithischer Architektur-Ansatz. Verwendung von Microservices
 - 9.6.2. Microservices-basierte Architekturen
 - 9.6.3. Kommunikationsflüsse bei der Verwendung von Microservices
- 9.7. Client-seitiger Aufruf von APIs
 - 9.7.1. Web-Client-Typologien
 - 9.7.2. Entwicklungswerkzeuge für die Verarbeitung von Webdiensten
 - 9.7.3. Ursprungsübergreifende Ressourcen (CORS)
- 9.8. Sicherheit von API-Aufrufen
 - 9.8.1. Sicherheit in Webdiensten
 - 9.8.2. Authentifizierung und Autorisierung
 - 9.8.3. Authentifizierungsmethoden auf Basis der Sicherheitsstufe
- 9.9. Integration von Anwendungen mit *Cloud*-Anbietern
 - 9.9.1. Anbieter des *Cloud Computing*
 - 9.9.2. Plattform-Dienste
 - 9.9.3. Webdienste Implementierung/Verbrauchsorientierte Dienste
- 9.10. Implementierung von *Bots* und Assistenten
 - 9.10.1. Verwendung von *Bots*
 - 9.10.2. Verwendung von Web Service in *Bots*
 - 9.10.3. Implementierung von *Chatbots* und Web-Assistenten

Modul 10. Programmierung in der Cloud. Projektmanagement und Produktüberprüfung

- 10.1. Kaskaden-Methoden
 - 10.1.1. Klassifizierung der Methodologien
 - 10.1.2. Kaskadenmodell. *Waterfall*
 - 10.1.3. *Strong and Weakness*
 - 10.1.4. Vergleich der Modelle. *Waterfall vs. Agile*
- 10.2. *Agile*-Methodik
 - 10.2.1. *Agile*-Methodik
 - 10.2.2. Das *Agile* Manifest
 - 10.2.3. Verwendung von *Agile*
- 10.3. Scrum-Methodik
 - 10.3.1. Scrum-Methodik
 - 10.3.1.1. Verwendung von Scrum
 - 10.3.2. Scrum-Events
 - 10.3.3. Scrum-Artefakte
 - 10.3.4. Scrum-Leitfaden
- 10.4. *Agile Inception Desk*
 - 10.4.1. *Agile Inception Desk*
 - 10.4.2. Phasen am *Inception Desk*
- 10.5. Technik des *Impact Mapping*
 - 10.5.1. *Impact Mapping*
 - 10.5.2. Nutzung des *Impact Mapping*
 - 10.5.3. Struktur des *Impact Mapping*
- 10.6. Anwenderberichte
 - 10.6.1. Anwenderberichte
 - 10.6.2. Schreiben von Anwenderberichten
 - 10.6.3. Hierarchie der Benutzergeschichten
 - 10.6.4. *Use Story Mapping*
- 10.7. Test Qa Manual
 - 10.7.1. *Testing Manual*
 - 10.7.2. Validierung und Verifizierung. Unterschiede
 - 10.7.3. Manuelles Testen. Typologie
 - 10.7.4. UAT. *User Acceptance Testing*
 - 10.7.5. UAT und Alpha- und Beta-Tests
 - 10.7.6. Software-Qualität
- 10.8. Automatisiertes Testen
 - 10.8.1. Automatisiertes Testen
 - 10.8.2. Manuelles vs. Automatisches Testen
 - 10.8.3. Die Auswirkungen des automatischen Testens
 - 10.8.4. Das Ergebnis der Anwendung von Automatisierung
 - 10.8.5. Das Qualitätsrad
- 10.9. Funktionale und nichtfunktionale Tests
 - 10.9.1. Funktionale und nichtfunktionale Tests
 - 10.9.2. Funktionsprüfung
 - 10.9.2.1. Einheitstests
 - 10.9.2.2. Integrationstests
 - 10.9.2.3. Regressionstests
 - 10.9.2.4. *Smoke Test*
 - 10.9.2.5. *Monkey Testing*
 - 10.9.2.6. *Sanity Testing*
 - 10.9.3. Nichtfunktionale Tests
 - 10.9.3.1. Belastungstests
 - 10.9.3.2. Leistungstests
 - 10.9.3.3. Sicherheitstests
 - 10.9.3.4. Testen der Konfiguration
 - 10.9.3.5. Stresstest
- 10.10. Verifizierungsmethoden und Tools
 - 10.10.1. *Heatmap*
 - 10.10.2. *Eye Tracking*
 - 10.10.3. *Scroll-Maps*
 - 10.10.4. Bewegungskarten
 - 10.10.5. *Konfetti-Maps*
 - 10.10.6. Test A/B
 - 10.10.7. *Blue & Green Deployment*-Methode
 - 10.10.8. *Canary Release*-Methode
 - 10.10.9. Auswahl der Tools
 - 10.10.10. Analytische Instrumente



“

Seien Sie ein echter Profi. Verringern Sie die Risiken in der Cloud und garantieren Sie die Sicherheit der Unternehmen, für die Sie arbeiten"

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 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.



07

Qualifizierung

Der Privater Masterstudiengang in Programmierung in der Cloud garantiert neben der präzisesten und aktuellsten Fortbildung 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"*

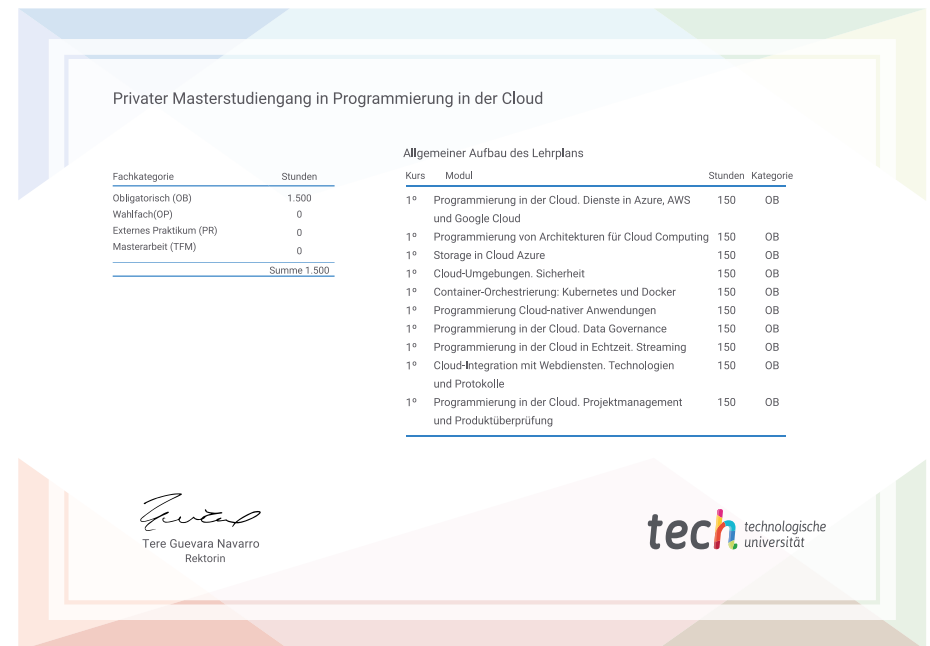
Dieser **Privater Masterstudiengang in Programmierung in der Cloud** enthält das vollständigste und aktuellste Programm auf dem Markt.

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 Programmierung in der Cloud**

Anzahl der offiziellen Arbeitsstunden: **1.500 Std.**



zukunft

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

tech technologische
universität

Privater Masterstudiengang

Programmierung in der Cloud

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

Privater Masterstudiengang Programmierung in der Cloud