

# Universitätsexperte

Anwendung von Techniken  
der Künstlichen Intelligenz  
im Lebenszyklus von  
Softwareprojekten

## Universitätsexperte

Anwendung von Techniken  
der Künstlichen Intelligenz  
im Lebenszyklus von Softwareprojekten

- » Modalität: **online**
- » Dauer: **6 Monate**
- » Qualifizierung: **TECH Technologische Universität**
- » Aufwand: **16 Std./Woche**
- » Zeitplan: **in Ihrem eigenen Tempo**
- » Prüfungen: **online**

# Index

01

Präsentation

---

Seite 4

02

Ziele

---

Seite 8

03

Kursleitung

---

Seite 12

04

Struktur und Inhalt

---

Seite 16

05

Methodik

---

Seite 24

06

Qualifizierung

---

Seite 32

# 01

# Präsentation

Die Softwareentwicklung von Webanwendungen, die künstliche Intelligenz (KI) nutzen, muss sich durch Sicherheit auszeichnen. Die Programmierer müssen die Privatsphäre der Nutzer schützen, die Integrität ihrer persönlichen Daten gewährleisten und die auf internationaler Ebene festgelegten Vorschriften einhalten. Immer mehr Unternehmen sind sich der Bedeutung guter Sicherheitspraktiken bei digitalen Projekten bewusst und fordern die Einbeziehung von IT-Experten in diesem Bereich. Auf diese Weise entwickeln die Institutionen Techniken zur Verhinderung von Cyber-Angriffen, wie z. B. SQL-Injection. Um den Fachleuten zu helfen, diese Möglichkeiten zu nutzen, hat TECH ein fortschrittliches 100% Online-Programm entwickelt, das es ihnen ermöglicht, sich mit der Software-Architektur für QA-Tests vertraut zu machen.



“

*Das System zum Wiederholen von Lerninhalten wird die langen Studienzeiten verkürzen, die bei anderen Lehrmethoden häufig auftreten“*

Werkzeuge der künstlichen Intelligenz spielen eine wichtige Rolle bei der Verbesserung der Produktivität sowohl in der Programmierung als auch in der Softwareentwicklung. Sie können beispielsweise Fehler effizienter erkennen und beheben und so den Zeitaufwand für die manuelle Fehlersuche verringern. Gleichzeitig suchen solche Mechanismen nach Sicherheitslücken und führen umfassende Sicherheitstests durch, wodurch der Schutz von Anwendungen optimiert wird. IT-Spezialisten können so Aspekte wie die Vorhersage von Fristen oder die Zuweisung von Ressourcen berücksichtigen, um ihre Planung zu verbessern.

In diesem Zusammenhang hat TECH eine bahnbrechende Fortbildung entwickelt, die Strategien zur Verbesserung der Produktivität bei der Softwareentwicklung mit künstlicher Intelligenz vermittelt. Auf der Tagesordnung stehen daher Aspekte wie die Verwaltung von Repositories, die Integration von maschinellem Lernen mit Datenbanken und die automatische Übersetzung zwischen Programmiersprachen.

Ein weiterer Schwerpunkt wird die Umsetzung von *Clean Architecture* in Informatikprozessen sein, da diese die Codequalität verbessert und eine kollaborative Entwicklung ermöglicht. Auf der anderen Seite liefert das Material die Grundlagen für die Erstellung von Projekten mit *Intelligent Computing*, sowohl in LAMP- als auch in MEVN-Umgebungen. Darüber hinaus werden mehrere reale Fallstudien und Übungen integriert, um die Programmentwicklung näher an die alltägliche Informatikpraxis heranzuführen.

Der Lehrplan basiert auf einer theoretisch-praktischen Perspektive und bietet Fachleuten ein intensives Lernen über Web-Projekte mit künstlicher Intelligenz. Die Studenten eignen sich die Inhalte mit Hilfe von Videozusammenfassungen zu jedem Thema, Fachliteratur und Infografiken an. Dank des *Relearning*-Systems von TECH werden die Programmierer auf natürliche Weise Fortschritte machen, neue Konzepte leichter aufnehmen und so die langen Lernzeiten verkürzen. Die einzige Voraussetzung für diesen Universitätsabschluss ist ein elektronisches Gerät mit Internetanschluss, um jederzeit auf den virtuellen Campus zugreifen zu können.

Dieser **Universitätsexperte in Anwendung von Techniken der Künstlichen Intelligenz im Lebenszyklus von Softwareprojekten** enthält das vollständigste und aktuellste Programm auf dem Markt. Die hervorstechendsten Merkmale sind:

- Die Entwicklung von Fallstudien, die von Experten für die Anwendung von Techniken der künstlichen Intelligenz im Lebenszyklus von Softwareprojekten vorgestellt werden
- Der anschauliche, schematische und äußerst praxisnahe Inhalt vermittelt alle für die berufliche Praxis unverzichtbaren wissenschaftlichen und praktischen Informationen
- 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



*Sie werden sich mit verschiedenen Strategien beschäftigen, die Ihnen bei der Wartbarkeit von Anwendungen mit Machine Learning helfen werden“*

“

*Sie werden sich mit der Code-Optimierung mit ChatGPT beschäftigen, einem der neuesten Trends, der die IT-Landschaft revolutioniert hat"*

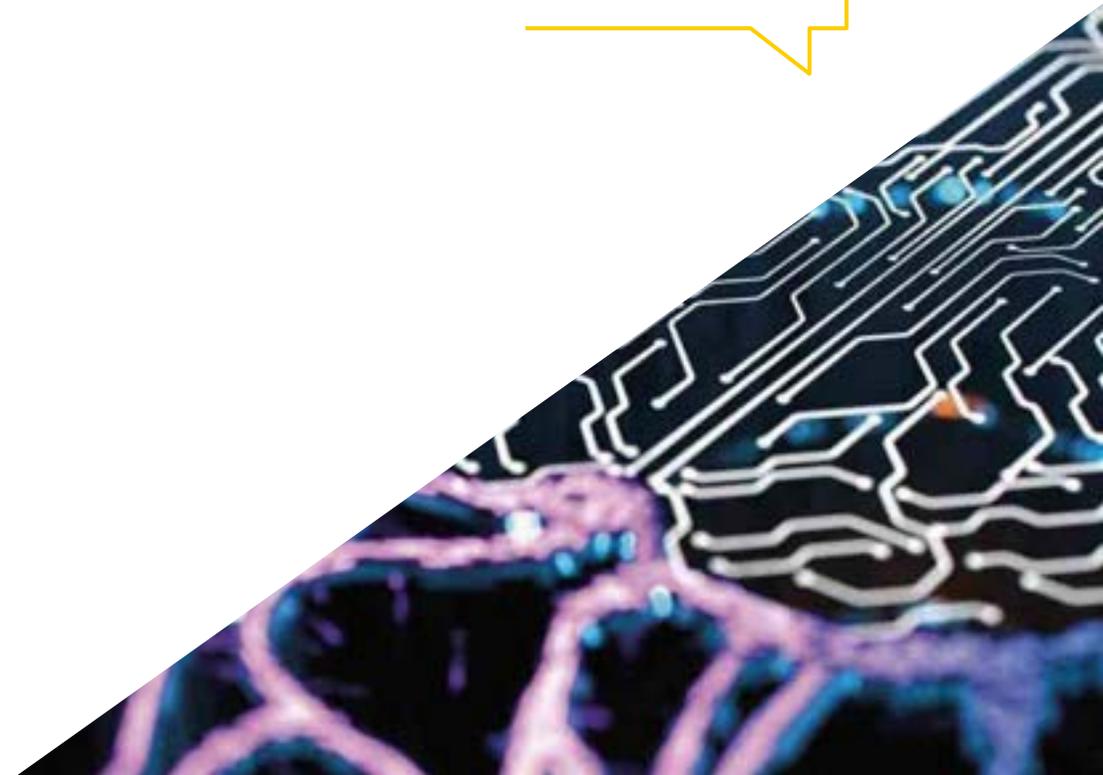
Zu den Dozenten des Programms gehören Experten aus der Branche, die ihre Berufserfahrung in diese Fortbildung einbringen, sowie renommierte Fachleute von Referenzgesellschaften und angesehenen Universitäten.

Die multimedialen Inhalte, die mit der neuesten Bildungstechnologie entwickelt wurden, werden der Fachkraft ein situiertes 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 Programms konzentriert sich auf problemorientiertes Lernen, bei dem die Fachkraft versuchen muss, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des gesamten Studiengangs gestellt werden. Zu diesem Zweck wird sie von einem innovativen interaktiven Videosystem unterstützt, das von renommierten Experten entwickelt wurde.

*Mit diesem innovativen 100%igen Online-Programm bereiten Sie die optimalen Entwicklungsumgebungen für Ihre IT-Prozesse vor.*

*Mit den didaktischen Werkzeugen von TECH, darunter erklärende Videos und interaktive Zusammenfassungen, werden Sie Ihre Ziele erreichen.*



# 02 Ziele

Nach Abschluss dieser Fortbildung beherrschen die Programmierer die Konfiguration von Softwareentwicklungsumgebungen, die durch den Einsatz von künstlicher Intelligenz optimiert wurden. Sie werden außerdem Visual Studio Code-Erweiterungen in ihre Arbeitsabläufe implementieren und so die Produktivität steigern. In diesem Zusammenhang entwickeln die Fachleute auch die Fähigkeit, solide Testpläne zu entwerfen, die verschiedene Arten von *Testing* umfassen, um die Qualität der Anwendungen zu gewährleisten. Darüber hinaus werden sie die fortschrittlichsten QA-Tools für eine optimale *Bug*-Erkennung einsetzen.



“

*Sie werden das Front-End entwickeln  
und hochgradig personalisierte  
Benutzererfahrungen entwerfen”*



## Allgemeine Ziele

---

- Entwickeln von Fähigkeiten zur Einrichtung und Verwaltung effizienter Entwicklungsumgebungen, um eine solide Grundlage für die Umsetzung von Projekten im Bereich der künstlichen Intelligenz zu schaffen
- Erwerben von Kenntnissen über die Planung, Durchführung und Automatisierung von Qualitätstests unter Einbeziehung von Tools der künstlichen Intelligenz zur Erkennung und Korrektur von *Bugs*
- Verstehen und Anwenden von Grundsätzen der Leistung, Skalierbarkeit und Wartbarkeit bei der Entwicklung von Großrechnersystemen
- Kennenlernen der wichtigsten Entwurfsmuster und deren effektive Anwendung in der Softwarearchitektur



*Sie werden das Frontend entwickeln  
und hochgradig personalisierte  
Benutzererfahrungen entwerfen"*





## Spezifische Ziele

---

### Modul 1. Produktivitätssteigerung in der Softwareentwicklung mit künstlicher Intelligenz

- ♦ Eintauchen in die Implementierung von unverzichtbaren Erweiterungen der künstlichen Intelligenz in Visual Studio Code, um die Produktivität zu steigern und die Softwareentwicklung zu erleichtern
- ♦ Gewinnen eines soliden Verständnisses grundlegender Konzepte der künstlichen Intelligenz und ihrer Anwendung in der Softwareentwicklung, einschließlich Algorithmen für maschinelles Lernen, Verarbeitung natürlicher Sprache, neuronale Netze usw.
- ♦ Beherrschen der Konfiguration optimierter Entwicklungsumgebungen, um sicherzustellen, dass die Studenten Umgebungen schaffen können, die für Projekte der künstlichen Intelligenz förderlich sind
- ♦ Anwenden spezifischer Techniken unter Verwendung von ChatGPT für die automatische Identifizierung und Korrektur potenzieller Code-Verbesserungen, wodurch effizientere Programmierpraktiken gefördert werden
- ♦ Fördern der Zusammenarbeit zwischen verschiedenen Programmierern (von Programmierern über Dateningenieure bis hin zu Designern für Benutzererfahrungen), um effektive und ethische Softwarelösungen künstlicher Intelligenz zu entwickeln

### Modul 2. Software-Architektur für QA-Testing

- ♦ Entwickeln von Fähigkeiten zur Erstellung robuster Testpläne, die verschiedene Arten von *Testing* abdecken und die Softwarequalität sicherstellen
- ♦ Erkennen und Analysieren verschiedener Arten von Softwarearchitekturen, wie monolithische, Microservices oder serviceorientierte
- ♦ Gewinnen eines umfassenden Überblicks über die Prinzipien und Techniken zur Entwicklung von Computersystemen, die skalierbar sind und große Datenmengen verarbeiten können

- ♦ Anwenden fortgeschrittener Fähigkeiten bei der Implementierung von künstlicher Intelligenz gestützter Datenstrukturen, um die Leistung und Effizienz von Software zu optimieren
- ♦ Entwickeln sicherer Entwicklungspraktiken, wobei der Schwerpunkt auf der Vermeidung von Schwachstellen liegt, um die Software-Sicherheit auf Architekturebene zu gewährleisten

### Modul 3. Künstliche Intelligenz für QA-Testing

- ♦ Beherrschen von Prinzipien und Techniken zur Entwicklung von Computersystemen, die skalierbar sind und große Datenmengen verarbeiten können
- ♦ Anwenden fortgeschrittener Fähigkeiten bei der Implementierung von Datenstrukturen, die durch künstliche Intelligenz gestützt sind, um die Leistung und Effizienz von Software zu optimieren
- ♦ Verstehen und Anwenden sicherer Entwicklungspraktiken mit Schwerpunkt auf der Vermeidung von Schwachstellen wie Injektion, um Softwaresicherheit auf der Architekturebene zu gewährleisten
- ♦ Erstellen automatisierter Tests, insbesondere in Web- und Mobilumgebungen, unter Einbeziehung von Tools der künstlichen Intelligenz zur Verbesserung der Effizienz des Prozesses
- ♦ Einsetzen fortschrittlicher durch künstliche Intelligenz gestützter QA-Tools für eine effizientere Erkennung von *Bugs* und kontinuierliche Softwareverbesserung

# 03

## Kursleitung

Die grundlegende Prämisse von TECH ist es, den Studenten eine hervorragende Weiterbildung zu bieten. Aus diesem Grund hat sie die Dozenten, die an diesem Programm teilnehmen, sorgfältig ausgewählt. Diese Fachleute verfügen über umfangreiche Berufserfahrung, da sie in renommierten technologischen Einrichtungen gearbeitet haben und jahrelang in der Forschung tätig waren. Dank des Wissens, das sie in die Lehrmaterialien einfließen lassen, können die Studenten ihr Verständnis erweitern und gleichzeitig neue Kompetenzen entwickeln, die sie in ihrer IT-Praxis anwenden können.



“

*Die vielfältigen Talente und das Fachwissen der Dozenten schaffen eine dynamische Lernumgebung. Lernen Sie mit den Besten!”*

## Leitung



### Dr. Peralta Martín-Palomino, Arturo

- ♦ CEO und CTO bei Prometheus Global Solutions
- ♦ CTO bei Korporate Technologies
- ♦ CTO bei AI Shepherds GmbH
- ♦ Berater und strategischer Unternehmensberater bei Alliance Medical
- ♦ Direktor für Design und Entwicklung bei DocPath
- ♦ Promotion in Computertechnik an der Universität von Castilla La Mancha
- ♦ Promotion in Wirtschaftswissenschaften, Unternehmen und Finanzen an der Universität Camilo José Cela
- ♦ Promotion in Psychologie an der Universität von Castilla La Mancha
- ♦ Masterstudiengang Executive MBA von der Universität Isabel I
- ♦ Masterstudiengang in Business und Marketing Management von der Universität Isabel I
- ♦ Masterstudiengang in Big Data bei Formación Hadoop
- ♦ Masterstudiengang in fortgeschrittener Informationstechnologie von der Universität von Castilla La Mancha
- ♦ Mitglied von: Forschungsgruppe SMILE



### Hr. Castellanos Herreros, Ricardo

- ◆ Spezialist für Computersystemtechnik
- ◆ *Chief Technology Officer* bei OWQLO
- ◆ Freelance Technischer Berater
- ◆ Entwickler von mobilen Anwendungen für eDreams, Fnac, Air Europa, Bankia, Cetelem, Banco Santander, Santillana, Groupón und Grupo Planeta
- ◆ Webentwickler für Openbank und Banco Santander
- ◆ Kurs in *Machine Learning Engineer* von Udacity
- ◆ Technischer Ingenieur für Computersysteme von der Universität von Castilla La Mancha

# 04

## Struktur und Inhalt

Dieser akademische Rundgang wird die Grundlagen für die Softwareentwicklung mit Hilfe von künstlicher Intelligenz sowie für die effiziente Verwaltung von Repositories aufzeigen. Zu diesem Zweck konzentriert sich der Kurs auf die *No-Code*-Gestaltung von Schnittstellen, die Übersetzung zwischen Programmiersprachen und den Einsatz intelligenter Werkzeuge zur Verbesserung der Produktivität von Softwareprogrammen. Die Speicherung großer Datenmengen wird ebenfalls eingehend analysiert, wobei fortgeschrittene Algorithmen und Strukturen behandelt werden. Darüber hinaus werden die Lehrmaterialien den Lebenszyklus des *Testing* beleuchten und den Studenten eine umfassende Sichtweise vermitteln, die sowohl die Effizienz als auch die Zuverlässigkeit von Produkten garantiert.

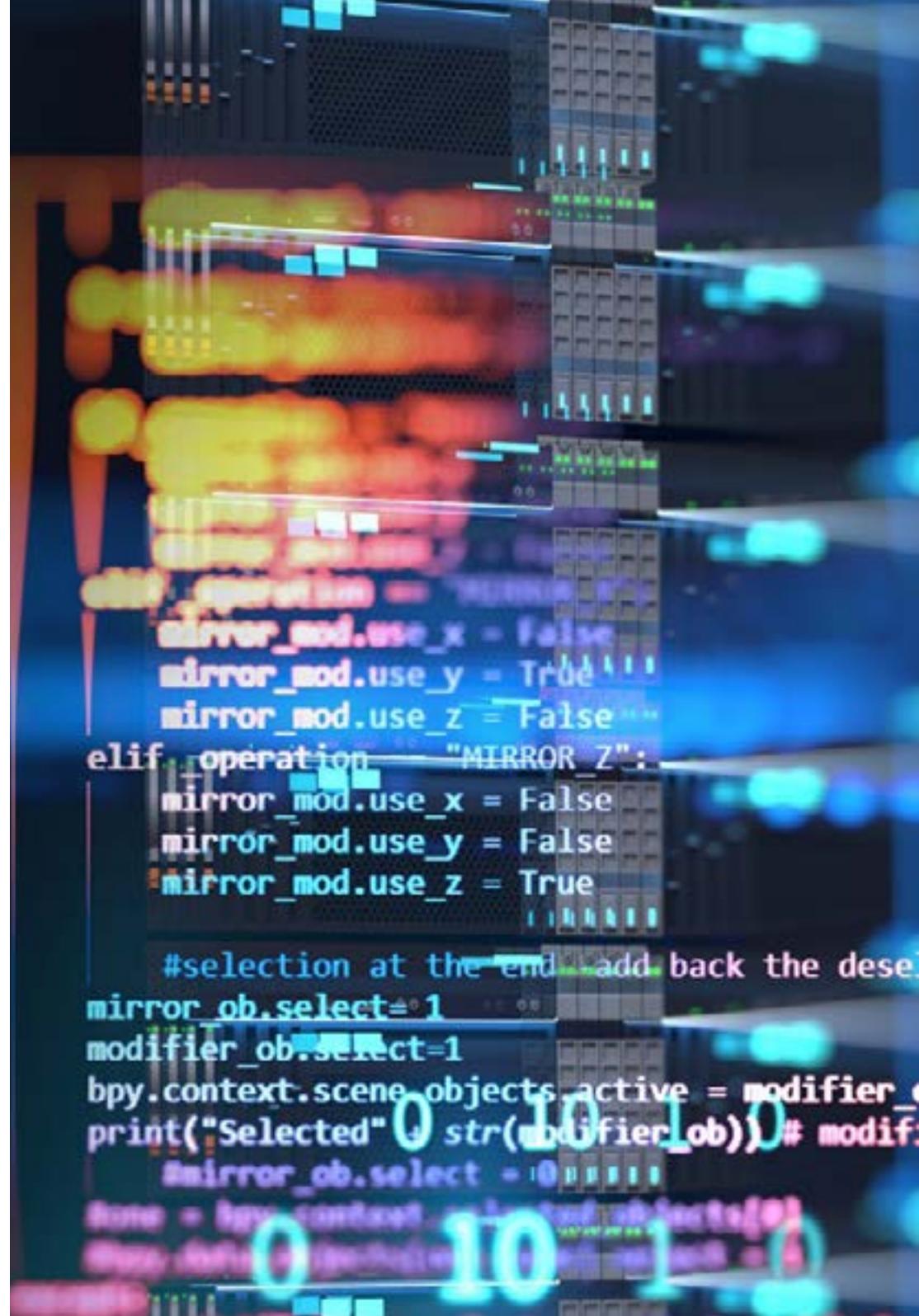


“

*Sie werden anhand von realen Fällen  
und der Lösung komplexer Situationen  
in simulierten Lernumgebungen  
arbeiten“*

## Modul 1. Produktivitätssteigerung in der Softwareentwicklung mit künstlicher Intelligenz

- 1.1. Vorbereiten einer geeigneten Entwicklungsumgebung
  - 1.1.1. Auswahl der wichtigsten Tools für die Entwicklung mit künstlicher Intelligenz
  - 1.1.2. Konfiguration der ausgewählten Tools
  - 1.1.3. Implementierung von CI/CD-Pipelines, die für Projekte mit künstlicher Intelligenz geeignet sind
  - 1.1.4. Effiziente Verwaltung von Abhängigkeiten und Versionen in Entwicklungsumgebungen
- 1.2. Wesentliche Erweiterungen mit künstlicher Intelligenz für Visual Studio Code
  - 1.2.1. Erkundung und Auswahl von Erweiterungen mit künstlicher Intelligenz für Visual Studio Code
  - 1.2.2. Integration von statischen und dynamischen Analysewerkzeugen in die IDE
  - 1.2.3. Automatisieren sich wiederholender Aufgaben mit spezifischen Erweiterungen
  - 1.2.4. Anpassung der Entwicklungsumgebung zur Verbesserung der Effizienz
- 1.3. No-Code-Design von Benutzeroberflächen mit Elementen künstlicher Intelligenz
  - 1.3.1. No-Code-Designprinzipien und ihre Anwendung auf Benutzeroberflächen
  - 1.3.2. Einbindung von Elementen künstlicher Intelligenz in das Design visueller Schnittstellen
  - 1.3.3. Tools und Plattformen für die No-Code-Erstellung von intelligenten Schnittstellen
  - 1.3.4. Bewertung und kontinuierliche Verbesserung von künstlicher Intelligenz gestützten No-Code-Schnittstellen
- 1.4. Code-Optimierung mit ChatGPT
  - 1.4.1. Identifizieren von doppeltem Code
  - 1.4.2. Refactoring
  - 1.4.3. Lesbaren Code erstellen
  - 1.4.4. Verstehen, was ein Code macht
  - 1.4.5. Verbesserung der Benennung von Variablen und Funktionen
  - 1.4.6. Automatische Dokumentation erstellen
- 1.5. Repository Management mit künstlicher Intelligenz
  - 1.5.1. Automatisierung von Versionskontrollprozessen mit Techniken künstlicher Intelligenz
  - 1.5.2. Konflikterkennung und automatische Lösung in kollaborativen Umgebungen
  - 1.5.3. Prädiktive Analyse von Änderungen und Trends in Code-Repositories
  - 1.5.4. Verbesserte Organisation und Kategorisierung von Repositories mithilfe von künstlicher Intelligenz



```
elif operation == "MIRROR_X":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end add back the deselected
mirror_ob.select=1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected", str(modifier_ob)) # modifier
#mirror_ob.select = 0
```

- 1.6. Integration künstlicher Intelligenz in die Datenbankverwaltung
  - 1.6.1. Abfrage- und Leistungsoptimierung durch Techniken künstlicher Intelligenz
  - 1.6.2. Prädiktive Analyse von Datenbankzugriffsmustern
  - 1.6.3. Implementierung von Empfehlungssystemen zur Optimierung der Datenbankstruktur
  - 1.6.4. Proaktive Überwachung und Erkennung von potenziellen Datenbankproblemen
- 1.7. Fehlersuche und Erstellung von *Unit-Tests* mit künstlicher Intelligenz
  - 1.7.1. Automatische Testfallerstellung mit Techniken künstlicher Intelligenz
  - 1.7.2. Frühzeitige Erkennung von Schwachstellen und *Bugs* durch statische Analyse mit künstlicher Intelligenz
  - 1.7.3. Verbesserung der Testabdeckung durch Identifizierung kritischer Bereiche mittels künstlicher Intelligenz
- 1.8. *Pair Programming* mit GitHub Copilot
  - 1.8.1. Integration und effektive Nutzung von GitHub Copilot in *Pair Programming*-Sitzungen
  - 1.8.2. Integration und verbesserte Kommunikation und Zusammenarbeit zwischen Entwicklern mit GitHub Copilot
  - 1.8.3. Integration von Strategien zur optimalen Nutzung der von GitHub Copilot generierten Code-Vorschläge
  - 1.8.4. Integration von Fallstudien und *Best Practices* in von künstlicher Intelligenz-unterstütztem *Pair Programming*
- 1.9. Automatische Übersetzung zwischen Programmiersprachen
  - 1.9.1. Programmiersprachenspezifische maschinelle Übersetzungstools und -dienste
  - 1.9.2. Anpassung von maschinellen Übersetzungsalgorithmen an den Entwicklungskontext
  - 1.9.3. Verbesserung der Interoperabilität zwischen verschiedenen Sprachen durch maschinelle Übersetzung
  - 1.9.4. Bewertung und Abschwächung potenzieller Herausforderungen und Einschränkungen bei der maschinellen Übersetzung
- 1.10. Empfohlene Tools künstlicher Intelligenz zur Verbesserung der Produktivität
  - 1.10.1. Vergleichende Analyse von Tools künstlicher Intelligenz für die Softwareentwicklung
  - 1.10.2. Integration von Tools künstlicher Intelligenz in Arbeitsabläufe
  - 1.10.3. Automatisierung von Routineaufgaben mit Tools künstlicher Intelligenz
  - 1.10.4. Bewertung und Auswahl von Tools auf der Grundlage von Projektkontext und Anforderungen

## Modul 2. Software-Architektur mit künstlicher Intelligenz

- 2.1. Optimierung und Leistungsmanagement in Tools künstlicher Intelligenz
  - 2.1.1. Leistungsanalyse und Profiling in Tools künstlicher Intelligenz
  - 2.1.2. Optimierungsstrategien für Algorithmen und -Modelle künstlicher Intelligenz
  - 2.1.3. Implementierung von *Caching*- und Parallelisierungstechniken zur Verbesserung der Leistung
  - 2.1.4. Tools und Methoden für die kontinuierliche Leistungsüberwachung in Echtzeit
- 2.2. Skalierbarkeit in Anwendungen der künstlichen Intelligenz
  - 2.2.1. Entwurf skalierbarer Architekturen für Anwendungen der künstlichen Intelligenz
  - 2.2.2. Implementierung von Partitionierungs- und Lastverteilungstechniken
  - 2.2.3. Workflow- und Workload-Management in skalierbaren Systemen
  - 2.2.4. Strategien für horizontale und vertikale Expansion in Umgebungen mit unterschiedlicher Nachfrage
- 2.3. Wartbarkeit von Anwendungen mit künstlicher Intelligenz
  - 2.3.1. Designprinzipien zur Erleichterung der Wartbarkeit in Projekten künstlicher Intelligenz
  - 2.3.2. Dokumentationsstrategien speziell für Modelle und -Algorithmen künstlicher Intelligenz
  - 2.3.3. Implementierung von Unit- und Integrationstests zur Vereinfachung der Wartung
  - 2.3.4. Methoden für *Refactoring* und kontinuierliche Verbesserung in Systemen mit Komponenten künstlicher Intelligenz
- 2.4. Entwurf von Großsystemen
  - 2.4.1. Architektonische Prinzipien für den Entwurf von Großsystemen
  - 2.4.2. Dekomposition komplexer Systeme in Mikrodienste
  - 2.4.3. Implementierung spezifischer Entwurfsmuster für verteilte Systeme
  - 2.4.4. Strategien zur Beherrschung der Komplexität in groß angelegten Architekturen mit Komponenten künstlicher Intelligenz
- 2.5. Großes *Data Warehousing* für Tools künstlicher Intelligenz
  - 2.5.1. Auswahl von skalierbaren Datenspeichertechnologien
  - 2.5.2. Entwurf von Datenbankschemata für die effiziente Handhabung großer Datenmengen
  - 2.5.3. Partitionierungs- und Replikationsstrategien in massiven Datenspeicherumgebungen
  - 2.5.4. Implementierung von Datenverwaltungssystemen zur Gewährleistung von Integrität und Verfügbarkeit in Projekten künstlicher Intelligenz

- 2.6. Datenstrukturen mit künstlicher Intelligenz
  - 2.6.1. Anpassung klassischer Datenstrukturen für die Verwendung in Algorithmen künstlicher Intelligenz
  - 2.6.2. Entwurf und Optimierung spezifischer Datenstrukturen für maschinelle Lernmodelle
  - 2.6.3. Integration von effizienten Datenstrukturen in datenintensive Systeme
  - 2.6.4. Strategien für Echtzeit-Datenmanipulation und Speicherung in Datenstrukturen künstlicher Intelligenz
- 2.7. Programmieralgorithmen für Produkte künstlicher Intelligenz
  - 2.7.1. Entwicklung und Implementierung von anwendungsspezifischen Algorithmen für Anwendungen künstlicher Intelligenz
  - 2.7.2. Algorithmenauswahlstrategien je nach Problemtyp und Produkthanforderungen
  - 2.7.3. Anpassung von klassischen Algorithmen für die Integration in Systeme künstlicher Intelligenz
  - 2.7.4. Bewertung und Vergleich der Leistung verschiedener Algorithmen in Entwicklungskontexten künstlicher Intelligenz
- 2.8. Entwurfsmuster für die Entwicklung künstlicher Intelligenz
  - 2.8.1. Identifizierung und Anwendung gemeinsamer Entwurfsmuster in Projekten mit Komponenten künstlicher Intelligenz
  - 2.8.2. Entwicklung von spezifischen Mustern für die Integration von Modellen und Algorithmen in bestehende Systeme
  - 2.8.3. Pattern-Implementierungsstrategien zur Verbesserung der Wiederverwendbarkeit und Wartbarkeit in Projekten künstlicher Intelligenz
  - 2.8.4. Fallstudien und *Best Practices* bei der Anwendung von Entwurfsmustern in Architekturen künstlicher Intelligenz
- 2.9. Implementierung einer *Clean Architecture*
  - 2.9.1. Grundlegende Prinzipien und Konzepte der *Clean Architecture*
  - 2.9.2. Anpassung der *Clean Architecture* an Projekte mit Komponenten künstlicher Intelligenz
  - 2.9.3. Implementierung von Schichten und Abhängigkeiten in *Clean Architecture* Systemen
  - 2.9.4. Vorteile und Herausforderungen der Implementierung von *Clean Architecture* in der Softwareentwicklung künstlicher Intelligenz



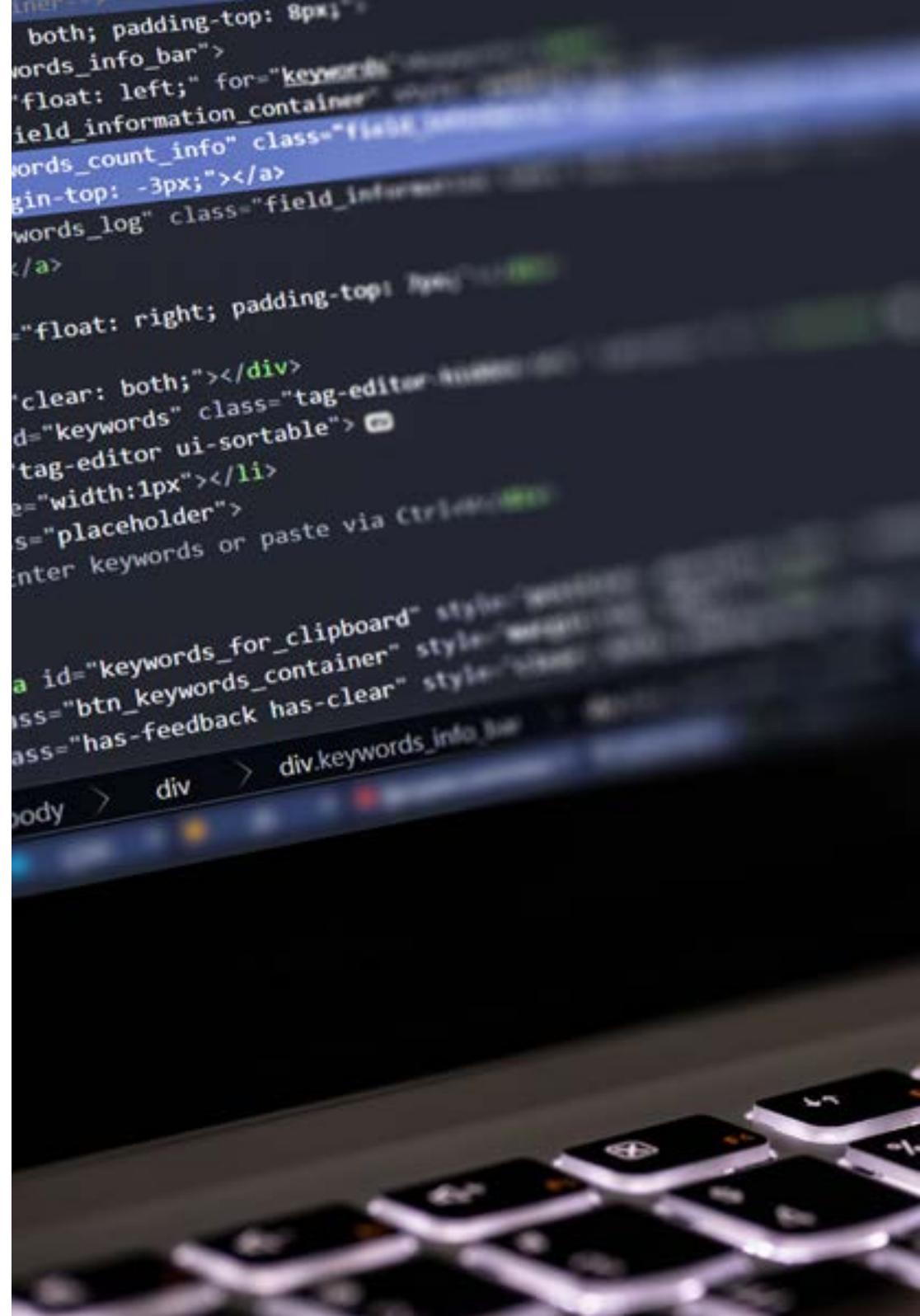


- 2.10. Sichere Softwareentwicklung in von künstlicher Intelligenz gestützten Webanwendungen
  - 2.10.1. Sicherheitsprinzipien bei der Softwareentwicklung mit Komponenten künstlicher Intelligenz
  - 2.10.2. Identifizierung und Entschärfung potenzieller Schwachstellen in Modellen und -Algorithmen künstlicher Intelligenz
  - 2.10.3. Implementierung von sicheren Entwicklungspraktiken in Webanwendungen mit Funktionalitäten künstlicher Intelligenz
  - 2.10.4. Strategien zum Schutz sensibler Daten und zur Verhinderung von Angriffen in Projekten künstlicher Intelligenz

**Modul 3. Künstliche Intelligenz für QA-Testing**

- 3.1. *Testing*-Lebenszyklus
  - 3.1.1. Beschreibung und Verständnis des *Testing*-Lebenszyklus in der Softwareentwicklung
  - 3.1.2. Phasen des *Testing*-Lebenszyklus und ihre Bedeutung für die Qualitätssicherung
  - 3.1.3. Integration von künstlicher Intelligenz in verschiedenen Phasen des *Testing*-Lebenszyklus
  - 3.1.4. Strategien zur kontinuierlichen Verbesserung des *Testing*-Lebenszyklus durch den Einsatz künstlicher Intelligenz
- 3.2. *Test Cases* und *Bug*-Erkennung
  - 3.2.1. Effektives Entwerfen und Schreiben von Testfällen im Kontext von QA *Testing*
  - 3.2.2. Identifizierung von *Bugs* und Fehlern während der Ausführung von Testfällen
  - 3.2.3. Anwendung von Techniken zur *Bugs*-Früherkennung durch statische Analyse
  - 3.2.4. Einsatz von Tools der künstlichen Intelligenz zur automatischen Identifizierung von *Bugs* in *Test Cases*
- 3.3. Arten von *Testing*
  - 3.3.1. Erkundung der verschiedenen *Testing*-Arten im Bereich der QS
  - 3.3.2. Unit-, Integrations-, Funktions- und Akzeptanztests: Merkmale und Anwendungen
  - 3.3.3. Strategien für die Auswahl und geeignete Kombination von *Testing*-Arten in Projekten künstlicher Intelligenz
  - 3.3.4. Anpassung konventioneller *Testing*-Arten an Projekte mit Komponenten der künstlichen Intelligenz
- 3.4. Erstellen eines Testplans
  - 3.4.1. Entwerfen und Strukturieren eines umfassenden Testplans
  - 3.4.2. Identifizierung von Anforderungen und Testszenarien in Projekten künstlicher Intelligenz
  - 3.4.3. Strategien für die manuelle und automatisierte Testplanung
  - 3.4.4. Bewertung und kontinuierliche Anpassung des Testplans entsprechend der Projektentwicklung

- 3.5. Erkennung und Meldung von *Bugs* mit künstlicher Intelligenz
  - 3.5.1. Implementierung automatischer *Bug*-Erkennungstechniken unter Verwendung von Algorithmen des maschinellen Lernens
  - 3.5.2. Einsatz von Tools der künstlichen Intelligenz für die dynamische Codeanalyse zur Suche nach potenziellen Fehlern
  - 3.5.3. Strategien für die automatische Erstellung von detaillierten Berichten über die von künstlicher Intelligenz entdeckten *Bugs*
  - 3.5.4. Effektive Zusammenarbeit zwischen Entwicklungs- und QA-Teams bei der Verwaltung von künstlicher Intelligenz-identifizierten *Bugs*
- 3.6. Automatisierte Tests mit künstlicher Intelligenz erstellen
  - 3.6.1. Entwicklung von automatisierten Testskripten für Projekte mit KI-Komponenten
  - 3.6.2. Integration von auf künstlicher Intelligenz-basierten Testautomatisierungstools
  - 3.6.3. Verwendung von Algorithmen des *Machine Learning* für die dynamische Generierung von automatisierten Testfällen
  - 3.6.4. Strategien für die effiziente Ausführung und Wartung von automatisierten Testfällen in Projekten künstlicher Intelligenz
- 3.7. *API Testing*
  - 3.7.1. Grundlegende Konzepte des *API-Testing* und seine Bedeutung in der QA
  - 3.7.2. Entwicklung von Tests zur Überprüfung von APIs in Umgebungen mit Komponenten der künstlichen Intelligenz
  - 3.7.3. Strategien zur Daten- und Ergebnisvalidierung bei *API-Testing* mit künstlicher Intelligenz
  - 3.7.4. Verwendung spezifischer Tools für *API-Testing* in Projekten künstlicher Intelligenz
- 3.8. Tools künstlicher Intelligenz für *Web-Testing*
  - 3.8.1. Erkundung von Tools künstlicher Intelligenz für die Testautomatisierung in Webumgebungen
  - 3.8.2. Integration von Technologien zur Aufgabenerkennung und visuellen Analyse in *Web-Testing*
  - 3.8.3. Strategien für die automatische Erkennung von Änderungen und Leistungsproblemen in Webanwendungen mithilfe künstlicher Intelligenz
  - 3.8.4. Bewertung spezifischer Tools zur Verbesserung der Effizienz von *Web-Testing* mit künstlicher Intelligenz



- 3.9. *Mobile Testing* mittels künstlicher Intelligenz
  - 3.9.1. Entwicklung von *Testing*-Strategien für mobile Anwendungen mit Komponenten künstlicher Intelligenz
  - 3.9.2. Integration spezifischer *Testing*-Werkzeuge für mobile Plattformen auf der Grundlage von künstlicher Intelligenz
  - 3.9.3. Einsatz von Algorithmen des maschinellen Lernens zur Erkennung von Problemen bei der Leistung mobiler Anwendungen
  - 3.9.4. Strategien für die Validierung spezifischer Schnittstellen und Funktionen mobiler Anwendungen mithilfe künstlicher Intelligenz
- 3.10. QA-Tools mit künstlicher Intelligenz
  - 3.10.1. Erkundung von QA-Tools und Plattformen mit KI-Funktionalität
  - 3.10.2. Bewertung von Tools für effizientes Testmanagement und Testdurchführung in Projekten künstlicher Intelligenz
  - 3.10.3. Einsatz von Algorithmen des *Machine Learning* für die Generierung und Optimierung von Testfällen
  - 3.10.4. Strategien für die effektive Auswahl und Einführung von QA-Tools gestützt von künstlicher Intelligenz

# 05

# 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.



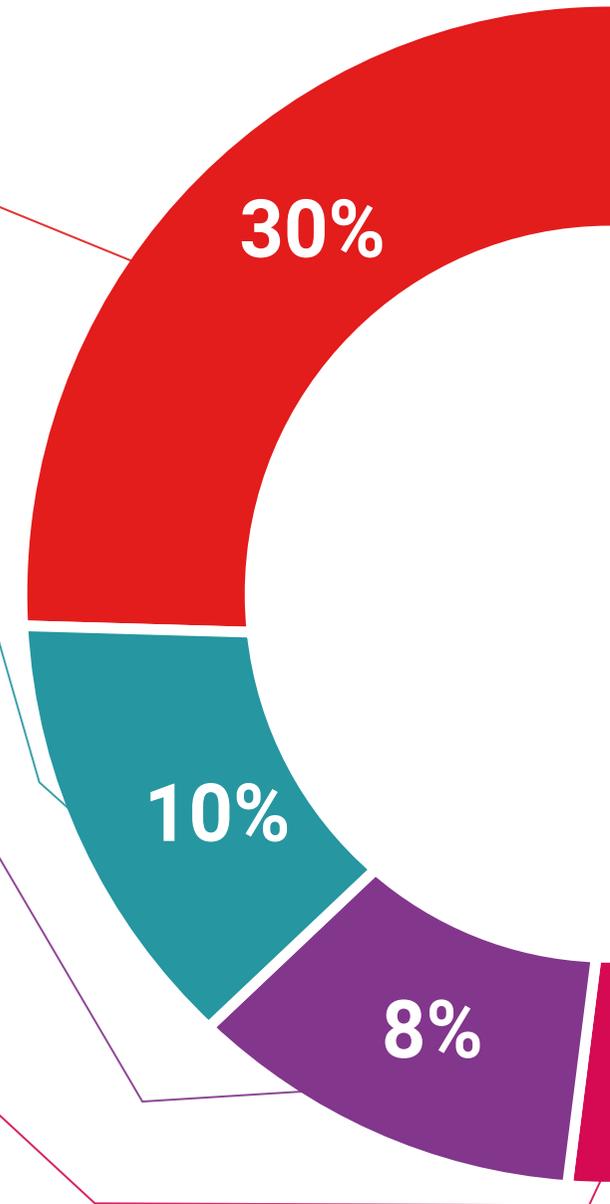
#### Übungen für Fertigkeiten und Kompetenzen

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.



06

# Qualifizierung

Der Universitätsexperte in Anwendung von Techniken der Künstlichen Intelligenz im Lebenszyklus von Softwareprojekten 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 **Universitätsexperte in Anwendung von Techniken der Künstlichen Intelligenz im Lebenszyklus von Softwareprojekten** 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: **Universitätsexperte in Anwendung von Techniken der Künstlichen Intelligenz im Lebenszyklus von Softwareprojekten**

Anzahl der offiziellen Arbeitsstunden: **450 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

**tech** technologische  
universität

### Universitätsexperte

Anwendung von Techniken der  
Künstlichen Intelligenz im Lebenszyklus  
von Softwareprojekten

- » Modalität: online
- » Dauer: 6 Monate
- » Qualifizierung: TECH Technologische Universität
- » Aufwand: 16 Std./Woche
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

# Universitätsexperte

Anwendung von Techniken  
der Künstlichen Intelligenz  
im Lebenszyklus von  
Softwareprojekten