

Privater Masterstudiengang

Anwendungsentwicklung für Android



Privater Masterstudiengang Anwendungsentwicklung für Android

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technische Universität
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Internetzugang: www.techtitude.com/de/informatik/masterstudiengang/masterstudiengang-anwendungsentwicklung-android

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

Heute gibt es weltweit mehr als 3 Milliarden Android-Geräte, und infolgedessen hat die Industrie für die Entwicklung mobiler Anwendungen ein rasantes Wachstum erlebt. Arbeitsmarktberichte bestätigen die wachsende Nachfrage nach Expertenprofilen im Bereich der Entwicklung mobiler Anwendungen, wobei Fachwissen über den gesamten Lebenszyklus der Anwendungsentwicklung, -bereitstellung und -vermarktung geschätzt wird. Es ist nicht leicht, Fachleute zu finden, die sich mit der Android-Programmiersprache, den Architekturen und Benutzeroberflächen auskennen. Aus diesem Grund wurde dieses Programm so konzipiert, dass es den Interessengruppen alle Kenntnisse vermittelt, um ihre Karriere mit einem spezialisierten Profil zu fördern. Sie können Ihren Abschluss vollständig online erwerben und in 12 Monaten erreichen.





“

Spezialisieren Sie sich auf die Schlüsselaspekte der Kundenerfahrung, um Android-Anwendungen so zu entwickeln, dass sie einen größeren Marktzugang haben"

Mit dem Privaten Masterstudiengang in Anwendungsentwicklung für Android können sich die Studenten auf eine der Standardtechnologien mit dem größten beruflichen Vorsprung dieser Zeit spezialisieren. Sie erwerben die notwendigen Kenntnisse, um eine Anwendung für Geräte mit diesem Betriebssystem vom Entwurf bis zur Produktion selbständig zu entwickeln. Außerdem erwerben sie spezielle Kenntnisse über den Lebenszyklus von Android-Anwendungen, um effizienter arbeiten zu können.

Zu diesem Zweck verfügt TECH über ein Team von hochqualifizierten Fachleuten, die über die notwendige Erfahrung verfügen, um reale Fälle zu vermitteln, die zu einem tieferen und effizienteren Lernprozess führen. Fachleute aus den Bereichen Programmierung, Anwendungsentwicklung und neue Technologien, die den Studenten im Studienprozess begleiten werden. Von einer sicheren Plattform aus, mit interaktiven Inhalten, die in Form von Videos, Grafiken, problemorientierten Beispielen und audiovisuellem Material verbreitet werden, und mit einer großen Gemeinschaft zur Debatte der aufgeworfenen Fragen.

So kann der Student in Bereichen wie IT oder Web oder Software, Geschäftsentwicklung, Marketing und Vertrieb, Beratung, plattformübergreifende Mobiltechnologie, Wirtschaft, Automobil, Heimautomatisierung, Internet der Dinge (IoT), Banken und Drohnen eingebunden werden. Und so sein Wissen zum Nutzen eines dieser Sektoren einsetzen.

Darüber hinaus verfügt TECH über die beste Studienmethodik, die auf *Relearning* basiert und vollständig online verfügbar ist, wodurch die Notwendigkeit, einen Klassenraum zu besuchen, oder die typischen Einschränkungen eines vorgegebenen Zeitplans entfallen. So erwerben die Studenten in 12 Monaten die notwendige Grundlage, um Beratungsaufgaben in allen Aspekten der mobilen Technologie auf Android zu übernehmen oder ein eigenes Unternehmen auf der Basis der Entwicklung von Anwendungen aufzubauen.

Dieser **Privater Masterstudiengang in Anwendungsentwicklung für Android** enthält das vollständigste und aktuellste Programm auf dem Markt. Die hervorstechendsten Merkmale sind:

- ◆ Die Entwicklung von Fallstudien, die von Experten für paralleles und verteiltes Rechnen vorgestellt werden
- ◆ Der anschauliche, schematische und äußerst praxisnahe Inhalt soll 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
- ◆ Ihr 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



Mit dem Privaten Masterstudiengang in Anwendungsentwicklung für Android werden Sie in der Lage sein, sich auf eine der Standardtechnologien mit dem größten beruflichen Vorsprung dieser Zeit zu spezialisieren



Sie erlangen die notwendigen Kenntnisse, um eine Anwendung für Android-Geräte vom Entwurf bis zur Produktion selbständig zu entwickeln"

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

Die multimedialen Inhalte, die mit den neuesten Bildungstechnologien entwickelt wurden, ermöglichen den Fachleuten ein situierendes 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 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.

Nutzen Sie die Vorteile des hochmodernen technologischen Umfelds, das Ihnen TECH bietet. Ein 100%iges Online-Programm, bei dem Sie mit führenden Experten arbeiten werden.

Er bietet Ihren Arbeitgebern oder Ihrem eigenen Unternehmen effiziente Lösungen an. Mit einer guten Handhabung der Grundlagen für die Entwicklung von Android-Anwendungen.

Android Studio
Version 3.3

Android Studio project

02 Ziele

Der Wachstumsprozess, der durch die unaufhaltsame Entwicklung neuer Technologien und Geräte ausgelöst wird, ist bekannt. Fortschritte, die eine ständige Fortbildung der IT- und Technikfachleute erfordern. Der Markt ist ständig in Bewegung, und um mit den Anforderungen Schritt zu halten, ist es für Fachleute unerlässlich, bei den Konzepten und Stilen, die gehandhabt werden, an vorderster Front zu stehen. Dieser Private Masterstudiengang in Anwendungsentwicklung für Android vermittelt alle notwendigen Hilfsmittel und Grundlagen, um sich in diesem Beruf weiterzuentwickeln und effektive Lösungen im beruflichen Umfeld anzubieten.



“

Setzen Sie Ihre eigenen Herausforderungen innerhalb des Ökosystems der Android-Technologie. Machen Sie erste Schritte auf dem Weg zu einer erfolgreichen Karriere als Anwendungsentwickler”



Allgemeine Ziele

- ◆ Bestimmung von strukturellen Elementen eines Android-Systems
- ◆ Analyse der Unterschiede zwischen den verschiedenen *Frameworks* für Entwicklung, Stärken und Schwächen
- ◆ Festlegen, wie Stile und Themen, einschließlich virtueller Realität, durch *Layouts* erstellt werden
- ◆ Analyse der verschiedenen Verwaltungsformen des Codeflusses in Kotlin
- ◆ Entwicklung von fortgeschrittenen Fähigkeiten und besten Praktiken in der Anwendungsentwicklung mit der Programmiersprache Kotlin
- ◆ Entwicklung einer Methodik für die optimale Datenverwaltung im Gerät
- ◆ Analyse der Anwendungsfälle für Android-Geräte auf dem Markt
- ◆ Beherrschen der Elemente des reaktionsfähigen Designs und Bewältigung der Herausforderungen eines solchen Designs
- ◆ Zusammenstellung der verschiedenen Phasen eines kontinuierlichen Integrationszyklus für die Android-Entwicklung



Werden Sie Experte für die Entwicklung von Android-Anwendungen. Melden Sie sich jetzt für dieses Programm an und machen Sie Ihren Abschluss zu 100% online in 12 Monaten"





Spezifische Ziele

Modul 1. Android-Programmiersprache

- ◆ Untersuchung des Linux-Kerns und der virtuellen Maschine auf der Grundlage von Android
- ◆ Analyse der systemeigenen Bibliotheken
- ◆ Festlegung der Vorteile von Android gegenüber anderen Plattformen
- ◆ Bestimmung der Elemente einer Android-Anwendung
- ◆ Vorstellung der Android-Versionen und ihrer Verbesserungen
- ◆ Bewertung des Android-Anwendungsmarktes
- ◆ Grundlagen für die zukünftige Entwicklung von Android

Modul 2. Verwendete *Frameworks* bei der Entwicklung von Android-Anwendungen

- ◆ Analysieren des Android Core *Framework*
- ◆ Entwicklung anderer *Frameworks*, die für die Entwicklung von Android-Anwendungen verwendet werden
- ◆ Implementierung von Libraries mit *Graddl*
- ◆ *Frameworks* für die Verbindung mit einer API einrichten
- ◆ Fachwissen über Architekturen wie MVP generieren
- ◆ Die Vor- und Nachteile von MVP und MVVM erfassen

Modul 3. Schnittstellen und *Layouts* in der Android-Anwendungsentwicklung

- ◆ Vorstellung des Lebenszyklusmodells von Android-Ansichten
- ◆ Untersuchung der wichtigsten Eigenschaften eines visuellen Designs (*Layouts*)
- ◆ Analyse der verfügbaren Designs (*Layouts*)
- ◆ Wiederverwendbares Design (*Layouts*) erstellen
- ◆ Bestimmen, wie alternative Ressourcen genutzt werden können
- ◆ Unterschiede in der Verwendung dieser Komponenten im Vergleich zu anderen Programmiersystemen identifizieren
- ◆ Festlegung des Potenzials und der Verwendung der Datei `AndroidManifest.xml`

Modul 4. Programmiersprache in Android-Anwendungen. Kotlin

- ◆ Entwicklung der Programmiersprache Kotlin
- ◆ Zusammenstellung der Funktionen und unterschiedlichen Möglichkeiten der Sprache Kotlin
- ◆ Untersuchung des grundlegenden Ausführungsmodells eines Programms in Kotlin
- ◆ Analyse der Sprachsyntax und der Struktur eines Programms in Kotlin
- ◆ Spezifizierung des Modells von Typen und Variablen in Kotlin
- ◆ Festlegung der verschiedenen Verwaltungsformen des Codeflusses in Kotlin
- ◆ Bestimmung des Modells von Klassen, Sammlungen und Objekten in Kotlin
- ◆ Schaffen von Fachwissen über das Vererbungsmodell in Kotlin
- ◆ Konkretisierung Verwaltungsmodell von Ausnahmen und Nulltypen in Kotlin

Modul 5. Programmiersprache in Android-Anwendungen. Fortgeschrittene Generizität von Kotlin, funktionale Programmierung und Parallelität

- ◆ Untersuchung des kovarianten und kontravarianten Generizitätsmodells in Kotlin
- ◆ Analyse des funktionalen Programmiermodells mit Kotlin Lambdas
- ◆ Spezifizierung der Funktionen höherer Ordnung von Kotlin
- ◆ Entwicklung der Erweiterungen und Begleitobjekte in Kotlin
- ◆ Untersuchung der Verwendung des Musters *Delegation* in Kotlin
- ◆ Zusammenstellung von Anmerkungen und Gedanken in Kotlin
- ◆ Vertiefung in das Modell *Testing* von Kotlin
- ◆ Festlegung der verschiedenen asynchronen Programmiermodelle, die in Kotlin verfügbar sind
- ◆ Bestimmung des Koroutinenmodells von Kotlin
- ◆ Zusammenstellung verschiedener Bibliotheken und Hilfsprogramme des Kotlin-Ökosystems

Modul 6. Datenverwaltung auf Android-Geräten

- ◆ Analyse der verschiedenen Techniken zur Datenverwaltung in Android
- ◆ Vorschlag von Methoden für die optimale Nutzung der Daten auf dem Gerät
- ◆ Identifizierung der für die Datenoptimierung erforderlichen Werkzeuge
- ◆ Untersuchung der Funktionen von JSON und XML für die Datenverwaltung unter Android
- ◆ Bewertung von allgemeinen Fragestellungen zu verteilten Systemen, die auf die Welt der Anwendungen für mobile Geräte anwendbar sind
- ◆ Bestimmung der Verwendung der Room-Bibliothek als Abstraktion für die Nutzung von SQLite auf Android sowie deren Vor- und Nachteile bestimmen
- ◆ Festlegung der erforderlichen Sicherheitsberechtigungen für die Datenverwaltung in jeder der in Android verwendeten Techniken

Modul 7. Tools für Android-Geräte

- ◆ Zusammenstellung der modernsten Tools für die tägliche Verwaltung
- ◆ Bewertung von Tools zur Steuerung von Android-Geräten
- ◆ Darstellung der Vorteile beim Einsatz von Android auf Drohnen
- ◆ Spezifizierung der Funktionalitäten von CRM-Tools
- ◆ Darstellung der Vorteile von Android-Geräten für Bankplattformen
- ◆ Analyse der Möglichkeiten der Plattform IoT und Android
- ◆ Untersuchung von Tools zur Prozesseffizienz auf Android

Modul 8. Design *Responsive* auf Android

- ◆ Analyse der Hauptelemente eines Designs
- ◆ Definition einer Designmethode für visuelle Elemente und einer Methode für das Screendesign
- ◆ Lösung verschiedener Probleme, die bei der Entwicklung mehrerer Geräte auftreten
- ◆ Über Tools verfügen, um bessere und mehr Ressourcen für die Entwicklung mehrerer Geräte zu generieren
- ◆ Untersuchung von alternative *Frameworks* zur nativen responsiven Entwicklung
- ◆ Schaffung einer eigenen Methodik für die Entwicklung von Anwendungen unter Verwendung bewährter Verfahren für die Anzeige auf mehreren Geräten von Beginn des Projekts an

Modul 9. Marketing für Android-Anwendungen

- ◆ Analyse von neuen nutzerzentrierte Methoden
- ◆ Bestimmen, wie die künstliche Intelligenz CX auf den nächsten Level gebracht hat
- ◆ Feststellung der Bedeutung von Zugänglichkeit und Mobilität
- ◆ Entwicklung modernster Techniken zur Sitzungs- und Verhaltensanalyse
- ◆ Spezifizierung von Mikropersonalisierungszielen während der Benutzer *Journey*
- ◆ Zusammenstellung neuer Methoden für ein sich wandelndes und lebendiges Umfeld zusammenstellen
- ◆ Vorschlag von Prototyping-Techniken

Modul 10. Lebenszyklus von Android-Anwendungen. Cloud, Playstore und Versionierung

- ◆ Spezifizierung der Vorteile eines automatisierten Modells für den Einsatz von Versionen
- ◆ Festlegung von Unterschieden zwischen kontinuierlicher Integration, Lieferung und Bereitstellung
- ◆ Definition der Hauptmerkmale von DevOps
- ◆ Bewertung einiger der grundlegenden Tools für die Implementierung von CI/CD-Pipelines
- ◆ Spezifizierung der wesentlichen Faktoren für die Entwicklung von Anwendungen, die CI/CD-Prozesse unterstützen
- ◆ Untersuchung von Containertechnologien als Grundpfeiler von CI/CD
- ◆ Ermittlung von Praktiken, Anwendungsfällen, Technologien und Tools aus dem CI/CD-Ökosystem

03 Kompetenzen

Es ist nicht einfach, Profile zu finden, die sich mit der Android-Programmiersprache, den Architekturen und Benutzeroberflächen auskennen, je nachdem, auf welchem Gerätetyp die Anwendung läuft. Mit dem Privaten Masterstudiengang in Anwendungsentwicklung für Android können sich die Studenten auf eine der Standardtechnologien mit dem größten beruflichen Vorsprung dieser Zeit spezialisieren. Sie erlangen die notwendigen Kenntnisse, um eine Anwendung für Android-Geräte vom Entwurf bis zur Produktion selbständig zu entwickeln. Außerdem erwerben spezielle Kenntnisse über den Lebenszyklus von Android-Anwendungen, um als Entwickler von Anwendungen für mobile Geräte effizienter arbeiten zu können.



“

*Mit diesem Privaten Masterstudiengang in
Anwendungsentwicklung für Android werden Sie Ihre
Karriere in der Wirtschaft oder in der Selbstständigkeit
vorantreiben. Mit Entwicklungen von Anwendungen,
die auf dem Markt erfolgreich sind"*



Allgemeine Kompetenzen

- ◆ Entwicklung von mobilen Anwendungen für Geräte mit Android-Betriebssystem
- ◆ Beherrschung der strukturellen Elementen eines Android-Systems
- ◆ Analyse der verschiedenen *Frameworks*, die für mehrere Architekturen verwendet werden sollen
- ◆ Definition der Eigenschaften einer Schnittstelle unter dem Gesichtspunkt der Benutzerfreundlichkeit
- ◆ Entwicklung von Fachwissen über die Programmiersprache Kotlin und den Kontext ihrer Entstehung
- ◆ Verwaltung des *Content Providers* für Datenaustausch und Datensicherheit
- ◆ Umgang mit Business-Tools auf Android: ihre Effizienz, Benutzerfreundlichkeit und Nützlichkeit
- ◆ Analyse der Elemente eines responsive Designs
- ◆ Analyse neuer nutzerzentrierter Methoden
- ◆ Anpassung der automatisierten Pipelines an das Android-Ökosystem

“

Wenn Sie all diese Kompetenzen in Ihrem Berufsprofil zusammenfassen, wird Ihr Angebot aufgewertet und Ihre Attraktivität auf dem Arbeitsmarkt erhöht“





Spezifische Kompetenzen

- ◆ Nutzung alternativer Ressourcen durch Analyse der verfügbaren Designs (*Layouts*)
- ◆ Entwicklung der verschiedenen Verwaltungsformen des Codeflusses in Kotlin und deren Fähigkeiten
- ◆ Verbesserung der Produktivität von Anwendungen durch die differenzierten Möglichkeiten der Sprache Kotlin
- ◆ Entwicklung der Erweiterungen und Begleitobjekte in Kotlin
- ◆ Verwendung von Datenbanken oder Netzwerkdiensten zur Erstellung von Dateien in verschiedenen Formaten
- ◆ Verwaltung der Room-Bibliothek als Abstraktion für die Verwendung von SQLite in Android
- ◆ Bewertung von Steuerungstools auf Android-Geräten, Analyse der Möglichkeiten von IoT und Android-Plattform
- ◆ Einführung von responsiv Design, um die Benutzerzufriedenheit zu erhöhen
- ◆ Kombination von Prototyping- und *Wireframing* -Techniken mit neuen bahnbrechenden Technologien wie künstlicher Intelligenz und dem Internet der Dinge

04

Kursleitung

Dieser Private Masterstudiengang in Anwendungsentwicklung für Android wird von Lehrkräften geleitet, die über umfassende Kenntnisse und Erfahrungen mit neuen Technologien, Lösungsarchitekturen und digitaler Infrastruktur verfügen und Experten für Android-Programmierung und Anwendungsentwickler sind. Ihre umfangreichen Lebensläufe garantieren die Qualität der für dieses Programm ausgewählten Inhalte. Sie setzen sich dafür ein, den Lernprozess für Fachleute zu optimieren, die in diesem Bereich den Beitrag suchen, den sie für ihren beruflichen Erfolg benötigen.



“

Professionelle Experten in der Entwicklung von Anwendungen und im technologischen Bereich werden Sie während des gesamten Studienprozesses begleiten und Sie mit den besten Inhalten versorgen”

Internationale Gastdirektorin

Colin Lee ist ein erfolgreicher Entwickler mobiler Anwendungen, der sich auf nativen Android-Code spezialisiert hat und dessen Einfluss sich international erstreckt. Der Experte ist eine Autorität in der Region Twin Cities und im Umgang mit Kotlin. Einer seiner jüngsten Beiträge bestand darin, in Live-Code zu demonstrieren, wie man mit der genannten Programmiersprache und den Open-Source-Browser-Komponenten von Mozilla für Android schnell einen Browser erstellen kann.

Darüber hinaus sind seine Anwendungen mit großen globalen Unternehmen verbunden. So war er zum Beispiel für die Entwicklung digitaler Lösungen für Pearson, einen der größten Verlage der Welt, verantwortlich. Er entwickelte auch einen Low-Level-Android-Videorekorder für das Startup Flipgrid, das später von Microsoft übernommen wurde.

Außerdem entwickelte er ein erfolgreiches Android-VPN für einen großen Beratungskunden. Er ist auch der Schöpfer eines Frachtmanagement-Tools, das von dem transnationalen Unternehmen Amazon eingesetzt wird, um die Arbeit seiner Vertragstrucker zu erleichtern. Außerdem hat er für Mozilla an der Entwicklung der mobilen Versionen des Firefox-Browsers mitgewirkt.

Heute ist er als Auftragnehmer tätig, unter anderem für Code-Reviews und Sicherheitsüberprüfungen. Sein Einfluss auf die Entwicklung mobiler Anwendungen und seine Erfahrung im Laufe der Jahre machen ihn zu einer führenden Persönlichkeit in der globalen Technologie-Arena.



Mr. Lee, Colin

- Direktor bei ColinTheShots LLC
- Android-Software-Ingenieur für Specto Inc.
- Leitender Android-Ingenieur für Mozilla
- Softwareentwicklungsingenieur für Amazon
- Ingenieur für mobile Anwendungen für Flipgrid
- Software-Konfigurationsspezialist für Pearson VUE
- Hochschulabschluss an der Universität von Florida

“

Eine einzigartige, wichtige und entscheidende Fortbildung, um Ihre berufliche Entwicklung zu fördern”

Leitung



Hr. Olalla Bonal, Martín

- ◆ *Actual Blockchain Technical Specialist bei IBM SPGI*
- ◆ Techniker für digitale Elektronik
- ◆ *Blockchain* Architekt
- ◆ Infrastruktur Architekt im Bankwesen
- ◆ Schulung *Hyperledger Fabric* für Unternehmen
- ◆ Geschäftsorientierte Schulung *Blockchain* für Unternehmen
- ◆ Projektleitung und Implementierung von Lösungen
- ◆ Mehr als 25 Jahre Erfahrung in der IT-Branche

Professoren

Hr. Villot Guisán, Pablo

- ◆ Cloud Architekt, Exponential Lösungen und Fachexperte *Blockchain* bei KPMG
- ◆ Entwickler und technischer Leiter von Web- und Desktop-Anwendungen für den Bereich Handelslogistik von Inditex, Connectis
- ◆ Hochschulabschluss in Computertechnik an der Universität La Coruña
- ◆ Microsoft MSCA-Zertifizierung: *Cloud Plattform*

Hr. Noguera Rodríguez, Pablo

- ◆ Entwickler für Native Apps (iOS & Android)- Starman Aviation (Aviaze App)
- ◆ Entwickler für Native Apps (iOS) - Stef (Mtrack App)
- ◆ Entwickler für Native Apps (iOS & Android)- Bitnovo (Bitnovo App)
- ◆ Entwicklungsexperte in Java: JSE, JEE und Android - Ilabora Bildung
- ◆ Programmierung von Android-Anwendungen - EOI - Madrid

Hr. Guerrero Díaz-Pintado, Arturo

- ◆ Berater für professionelle Dienstleistungen, der seit IBM mit führenden Organisationen in Europa, dem Nahen Osten und Lateinamerika zusammenarbeitet
- ◆ Herausragende Kooperationen mit renommierten Universitäten und Hochschulen in technologiebezogenen Themen wie Künstliche Intelligenz, Internet of Things, Cloud, Customer Experience und Digital Transformation
- ◆ Technischer Ingenieur für den Vertrieb des Watson Customer Engagement Portfolios (Marketing und Customer Experience Lösungen) in Spanien, Portugal, Griechenland und Israel bei IBM
- ◆ FuE-Netzwerkingenieur für FuE bei Telefónica
- ◆ Hochschulabschluss in Telekommunikationstechnik an der Universität Alcalá und der Danish Technical University

Hr. Pérez Rico, Javier

- ◆ Derzeitiger Technischer Leiter Android bei Nologis
- ◆ Technischer Leiter Android bei Seekle
- ◆ Android-Programmierer bei Gowex-Ideup
- ◆ Junior Android-Programmierer bei Tecnocom
- ◆ Referent des II Simposio iTest, E@tic2011
- ◆ Technischer Ingenieur für Computersysteme an der Universität Complutense in Madrid
- ◆ Masterstudiengang in Forschung an der Universität Complutense in Madrid

Hr. Jiménez Pérez, Carlos

- ◆ Senior Android-Entwickler bei OnTheSpot - Telefónica Tech
- ◆ Ingenieur für Automatisierungstechnik und Elektronik
- ◆ Masterstudiengang in Elektronischen Systemen und Anwendungstechnik
- ◆ Außerordentlicher Professor an der Universität Carlos III von Madrid

Hr. Marcano Van Grieken, Alejandro Antonio

- ◆ Produktmanager - Vikua, Remote (Jira, SCRUM, Figma, Slack, Notion)
- ◆ Backend-Entwickler - InnovativeGX
- ◆ Hochschulabschluss in Systemtechnik, Metropolitanische Universität von Caracas, Venezuela
- ◆ Masterstudiengang in Cybersicherheit, Universität von León Online

Hr. Arevalillo González, Emilio

- ◆ DBA Oracle BBVA
- ◆ *Assistant Project Manager Archibus Solution Center Spain*
- ◆ Backend developer bei Telefónica FuE
- ◆ Hochschulabschluss in Computertechnik an der Polytechnischen Universität von Madrid
- ◆ Masterstudiengang in Software und Systemen an der Polytechnischen Universität von Madrid

Hr. Arranz, Héctor

- ◆ Software-Projektleiter bei Ezenit
- ◆ Hochschulabschluss in Softwaretechnik an der Universität Complutense von Madrid
- ◆ MBA Power Leaders von The Power MBA Lehr-Erfahrung
- ◆ Ausbilder für digitale Fähigkeiten bei Three Life
- ◆ Ausbilderin für digitale Fähigkeiten bei Stiftung Esplai
- ◆ Außerordentlicher Professor für den Studiengang Multiplattform-Anwendungsentwicklung am MEDAC
- ◆ Unterstützung der unternehmerischen Tätigkeit an der Universität Complutense von Madrid
- ◆ Ausbilder für digitale Fähigkeiten bei Three Life y Fundacion Esplai
- ◆ Berater für Unternehmertum bei Cink Venturing
- ◆ Außerordentlicher Professor für den Studiengang Multiplattform-Anwendungsentwicklung am MEDAC

Hr. Gonzalo Fernández, Juan Luis

- ◆ Blockchain-basierter Produktmanager für Open Canarias
- ◆ Blockchain DevOps Direktor bei Alastria
- ◆ Direktor für Service Level Technologie bei Santander Spanien
- ◆ Direktor Mobile Anwendungsentwicklung Tinkerlink bei Cronos Telecom
- ◆ Technischer Direktor für IT-Service-Management bei der Barclays Bank Spanien
- ◆ Hochschulabschluss in Computertechnik an der UNED
- ◆ Spezialisierung auf Deep Learning bei DeepLearning.ai

05

Struktur und Inhalt

Dieses Programm wurde von Fachleuten entwickelt, die die aktuelle Realität und die zukünftigen Trends der Anwendungsentwicklung auf Android-Betriebssystemen kennen. Es gibt 10 Module, die Ihnen alles beibringen, was Sie brauchen, um ein Entwickler von Anwendungen zu werden. Von Modul 1 an sind die Technik und die vertieften Kenntnisse des Themas offensichtlich und können in 12 Monaten perfekt beherrscht werden. Erlangung einer 100%igen Online-Qualifikation dank der Methode *Relearning*, die ein schnelles Lernen und Einprägen der Konzepte und Praktiken ermöglicht.



“

Dynamische Inhalte mit praktischen Übungen, Videos und interaktiven Formaten sorgen für ein agiles und leicht verdauliches Training”

Modul 1. Android-Programmiersprache

- 1.1. Android-Plattform
 - 1.1.1. Android-Plattform
 - 1.1.2. Betriebssystem Android
 - 1.1.3. *Open Handset Alliance* in der Android-Entwicklung
- 1.2. Android-Architektur
 - 1.2.1. Architektonische Elemente eines Android-Systems
 - 1.2.2. Kommunikation zwischen den Elementen
 - 1.2.3. Erweiterbarkeit der Android-Architektur
 - 1.2.4. Verwaltung der Maschinenressourcen: Batterie und Speicher
 - 1.2.5. Android-Emulatoren
- 1.3. Android Linux-Kern
 - 1.3.1. Zusammensetzung des Kerns
 - 1.3.2. Strukturelle Elemente des Kerns
 - 1.3.3. Die virtuelle Dalvik-Maschine
 - 1.3.4. Die virtuelle Android-Laufzeitmaschine (ART)
- 1.4. Native Android-Bibliotheken
 - 1.4.1. Native Android-Bibliotheken
 - 1.4.2. Hilfsbibliotheken (*Support Library*)
 - 1.4.3. Native Bibliotheken und Erweiterbarkeit
- 1.5. Das Datei- und Dateisystem in Android
 - 1.5.1. Aufbau einer typischen Android-Anwendung
 - 1.5.2. YAFFS2 und ext4-Dateisystem
 - 1.5.3. Verwendung von SQLite und Room für die Datenverwaltung
- 1.6. Android-Sicherheit
 - 1.6.1. Genehmigungssystem
 - 1.6.2. Digitale Signaturen in den *Android Application Package* (apk)
 - 1.6.3. Laufende Prozesse im Kern
 - 1.6.4. Ausführungsfäden und Ereignisse
- 1.7. Strukturelle Komponenten einer Standardanwendung
 - 1.7.1. Ansicht (*View*)
 - 1.7.2. Aktivität (*Activity*)
 - 1.7.3. Fragment (*Fragment*)

- 1.7.4. Service (*Service*)
- 1.7.5. Absicht (*Intent*)
- 1.7.6. Broadcasts *Receiver* und *Content Provider*
- 1.7.7. Datenverwaltung und Nutzerpräferenzen
- 1.8. Android-Versionen
 - 1.8.1. Android-Versionen
 - 1.8.2. Einsatz von Android-Versionen
 - 1.8.3. Streuung der Android-Verteilungen
 - 1.8.4. Android vs. Apple iOS und andere mobile Betriebssysteme
- 1.9. Android für Fahrzeuge
 - 1.9.1. Android und die Automobilwelt
 - 1.9.2. Strukturelle Elemente in einem Android-System für Kraftfahrzeuge
 - 1.9.3. Kommunikation zwischen Geräten
- 1.10. Android in Domotik, *Wearables* und *Internet of Things* (IoT)
 - 1.10.1. Die vernetzte Welt
 - 1.10.2. Strukturelle Elemente in einem domotischen Android-System
 - 1.10.3. Elemente von *Android Wearable*
 - 1.10.4. Android in *Internet of Things* (IoT)

Modul 2. Bei der Entwicklung von Android-Anwendungen verwendete *Frameworks*

- 2.1. *Frameworks* in der Entwicklung von Android-Anwendungen
 - 2.1.1. Bei der Entwicklung von Android-Anwendungen verwendete *Frameworks*
 - 2.1.2. *Frameworks*. Typologien
 - 2.1.3. Wahl des *Frameworks* für das Projekt
- 2.2. Implementierung von *Frameworks* in Android
 - 2.2.1. *Android Frameworks Core* für Java/Kotlin
 - 2.2.2. *Jetpack Compose*
 - 2.2.3. *Frameworks* in anderen Sprachen
- 2.3. In der Entwicklung befindliche Bibliotheksverwaltungssysteme
 - 2.3.1. *Graddle*
 - 2.3.2. Automatisierung mit *Graddle*
 - 2.3.3. Entwicklungstool Maven

- 2.4. Reiner Code
 - 2.4.1. Angeordneter Code
 - 2.4.2. Code-Vorbereitung in Android-Anwendungen
 - 2.4.3. *Bikeshedding* und Priorisierung
- 2.5. Entwicklungsmuster in Android
 - 2.5.1. Muster-Kategorien
 - 2.5.2. Unterschiede zwischen Mustern
 - 2.5.3. *Factory, Observer und Singleton*
- 2.6. MVP. Modell, Ansicht und Darsteller
 - 2.6.1. MVC. Modell, Ansicht und Controller
 - 2.6.2. Modell, Ansicht und Darsteller
 - 2.6.3. Praktisches Beispiel: *Pokémon Battle*
- 2.7. MVVM. Modell, Ansicht und View Model
 - 2.7.1. MVC vs MVVM
 - 2.7.2. Modell, Ansicht und *View Model*
 - 2.7.3. Praktisches Beispiel: *Pokémon Battle II*
- 2.8. *Frameworks* und die am häufigsten in Android verwendeten Bibliotheken
 - 2.8.1. Interaktionsbibliotheken mit API
 - 2.8.2. Bibliotheken zur Datenkonvertierung
 - 2.8.3. *Firebase* und *Firebase Analytics*
- 2.9. *VisuellesFramework* für Android
 - 2.9.1. Lebenszyklus einer Android-Anwendung
 - 2.9.2. Entwurf von Ansichten in XML
 - 2.9.3. Entwurf von Elementen und Animationen in XML
- 2.10. *AndroidFrameworks* in anderen Sprachen
 - 2.10.1. *React Native*
 - 2.10.2. *Flutter*
 - 2.10.3. *Ionic*

Modul 3. Schnittstellen und *Layouts* in der Anwendungsentwicklung für Android

- 3.1. Lebenszyklus einer Android-Schnittstelle
 - 3.1.1. Lebenszyklus bei Android
 - 3.1.2. Verhältnis zwischen Prozess und Aktivität
 - 3.1.3. Persistenz des Status der Anwendung
 - 3.1.4. Architektur *Clean* für Android
- 3.2. Ansichten in der Entwicklung von Android-Anwendungen (*Views*)
 - 3.2.1. Architektur - Präsentationsschicht *Clean*
 - 3.2.2. *Recycler View*
 - 3.2.3. *Adapter View*
- 3.3. Designs in der Entwicklung von Android-Anwendungen (*Layouts*)
 - 3.3.1. *Layouts* in Android
 - 3.3.2. *Constraint Layout*
 - 3.3.3. Erstellen von *Layouts* mit dem *Android Studio Layout Editor*
- 3.4. Animationen in der Entwicklung von Android-Anwendungen (*Animationen*)
 - 3.4.1. Icons und Bilder
 - 3.4.2. Übergänge
 - 3.4.3. Unterschied zwischen Eigenschaftsanimation und Ansichtsanimation
- 3.5. Aktivitäten und Intentionen in der Entwicklung von Android-Anwendungen (*Activity und Intentions*)
 - 3.5.1. Explizite und implizite Absichten
 - 3.5.2. Aktionsleiste
 - 3.5.3. Kommunikation zwischen Aktivitäten
- 3.6. Alternative und Systemressourcen (*Material Design, Cardboard, etc.*)
 - 3.6.1. *Material Design* für Android
 - 3.6.2. Multimedia in Android
 - 3.6.3. Virtuelle Realität mit *Google Cardboard for Android NDK*
- 3.7. Stile und Themen in der Entwicklung von Android-Anwendungen
 - 3.7.1. Stile in einem Android-Projekt
 - 3.7.2. Themen für das Android-Projekt
 - 3.7.3. Wiederverwendung von Stilen und Themen
- 3.8. Grafiken, Touchscreen und Sensoren
 - 3.8.1. Arbeiten mit fortgeschrittenen Grafiken
 - 3.8.2. Verwaltung von Geräten mit Touchscreen und Tastatur
 - 3.8.3. Nutzung der Sensoren des Android-Geräts

- 3.9. Designs für erweiterte Realität
 - 3.9.1. Komplexe Schnittstellen mit der Kamera
 - 3.9.2. Positionssensoren und GPS in der erweiterten Realität
 - 3.9.3. Präsentation auf nicht standardisierten Displays
 - 3.9.4. Gängige Fehler und Probleme
- 3.10. Erweiterte Konfiguration der Benutzeroberfläche mit AndroidManifest.xml
 - 3.10.1. Die Leistungsfähigkeit der Android-Manifestdatei
 - 3.10.2. Programmatisches Design versus deklaratives Design
 - 3.10.3. Hauptbestandteile des Archivs

Modul 4. Programmiersprache in Android-Anwendungen. Kotlin

- 4.1. Programmiersprache Kotlin
 - 4.1.1. Programmiersprache Kotlin
 - 4.1.2. Programmiersprache Kotlin. Eigenschaften
 - 4.1.3. Ausführung eines Programms in Kotlin
- 4.2. Programmieren in Kotlin
 - 4.2.1. Struktur eines Kotlin-Programms
 - 4.2.2. Reservierte Wörter und Syntax
 - 4.2.3. Konsoleneingabe und Lesen von Benutzereingaben - Hello World
- 4.3. Typen und Variablen in Kotlin
 - 4.3.1. Typen und Variablen in Kotlin
 - 4.3.2. Erklärung der Variablen: Var vs. Val
 - 4.3.3. Betreiber
 - 4.3.4. Typ Konvertierung
 - 4.3.5. Arrays
- 4.4. Flusskontrolle in Kotlin
 - 4.4.1. Flusskontrolle
 - 4.4.2. Bedingte Ausdrücke
 - 4.4.3. Schleifen
- 4.5. Funktionen in Kotlin
 - 4.5.1. Funktionen in Kotlin
 - 4.5.2. Struktur einer Funktion
 - 4.5.3. Aufsuchende Funktionen (*Scope Funktionen*)





- 4.6. Klassen und Objekte in Kotlin
 - 4.6.1. Klassen und Objekte in Kotlin
 - 4.6.2. Klassen
 - 4.6.3. Objekte
 - 4.6.4. Konstruktoren und Eigenschaftsinitialisierung
 - 4.6.5. Verzweigte Klassen und innere Klassen
 - 4.6.6. Arten von Daten
- 4.7. Vererbung in Kotlin
 - 4.7.1. Vererbung
 - 4.7.2. Oberklassen und Unterklassen
 - 4.7.3. Überschreiben von Eigenschaften und Funktionen
 - 4.7.4. Vererbung vs. andere Arten von Beziehungen zwischen Klassen
 - 4.7.5. Versiegelte Klassen
 - 4.7.6. Aufzählung
- 4.8. Abstrakte Klassen und Benutzeroberflächen in Kotlin
 - 4.8.1. Abstrakte Klassen und Benutzeroberflächen
 - 4.8.2. Abstrakte Klassen
 - 4.8.3. Schnittstellen
 - 4.8.4. Typüberprüfung und -konvertierung - Operatoren Ist, Wenn, Als
- 4.9. Sammlungen in Kotlin
 - 4.9.1. Sammlungen in Kotlin
 - 4.9.2. Liste
 - 4.9.3. Satz
 - 4.9.4. Karte
- 4.10. Verwaltung von Ausnahmen und Nullwerten in Kotlin
 - 4.10.1. Verwaltung von Ausnahmen und Nullwerten
 - 4.10.2. Nullwert, Typen *Nullable* und *Non-Nullable*
 - 4.10.3. Ausnahmen

Modul 5. Programmiersprache in Android-Anwendungen. Fortgeschrittene Generizität von Kotlin, funktionale Programmierung und Parallelität

- 5.1. Generizität in Kotlin
 - 5.1.1. Generizität in Kotlin
 - 5.1.2. Generizität in Sammlungen, Funktionen, Klassen und Benutzeroberflächen
 - 5.1.3. Kovarianz und Kontravarianz: *Out* oder *In*
- 5.2. Funktionen *Lambda* in Kotlin
 - 5.2.1. Funktionen *Lambdas*
 - 5.2.2. Struktur einer Funktion *Lambda*
 - 5.2.3. Nutzung von Funktionen *Lambda*
- 5.3. Funktionen höherer Ordnung in Kotlin
 - 5.3.1. Funktionen höherer Ordnung
 - 5.3.2. Standard Funktionen höherer Ordnung in Kotlin
 - 5.3.3. Verknüpfung von Aufrufen an Funktionen
- 5.4. Erweiterungen in Kotlin
 - 5.4.1. Erweiterungen in Kotlin
 - 5.4.2. Erweiterungsfunktionen
 - 5.4.3. Eigenschaften der Erweiterung
 - 5.4.4. Begleitende Objekte
- 5.5. Muster *Delegation* in Kotlin
 - 5.5.1. Muster *Delegation*
 - 5.5.2. *Delegation* in Kotlin
 - 5.5.3. Delegierte Eigenschaften
- 5.6. Anmerkungen und Reflexion in Kotlin
 - 5.6.1. Anmerkungen und Reflexion
 - 5.6.2. Anmerkungen in Kotlin
 - 5.6.3. Reflexion in Kotlin
- 5.7. *Testing* in Kotlin
 - 5.7.1. *Testing* in Kotlin
 - 5.7.2. *Frameworks* und Bibliotheken von *Testing* in Kotlin
 - 5.7.3. Kotest

- 5.8. Asynchrone Programmierung in Kotlin
 - 5.8.1. Asynchrone Programmierung
 - 5.8.2. Asynchrone Programmierungstechniken in Kotlin
 - 5.8.3. Vergleich der Programmierungstechniken
- 5.9. Koroutinen in Kotlin
 - 5.9.1. Koroutinen
 - 5.9.2. Kanäle
 - 5.9.3. Kontext und *Dispatchers*
 - 5.9.4. Geteilter Stand und Konkurrenz
 - 5.9.5. Verwaltung von Ausnahmen in Koroutinen
- 5.10. Ökosystem Kotlin
 - 5.10.1. Ökosystem Kotlin
 - 5.10.2. Bibliotheken für Kotlin
 - 5.10.3. Tools für Kotlin

Modul 6. Datenverwaltung auf Android-Geräten

- 6.1. Datenverwaltung. Typologie
 - 6.1.1. Datenverwaltung auf mobilen Geräten
 - 6.1.2. Alternativen für die Datenverwaltung auf Android-Geräten
 - 6.1.3. Datengenerierung für die Arbeit mit künstlicher Intelligenz und Nutzungsanalyse
 - 6.1.4. Tools zur Leistungsmessung für eine optimale Datenverwaltung
- 6.2. Verwaltung der Benutzerpräferenz
 - 6.2.1. Arten von Daten, die in den Präferenzdateien enthalten sind
 - 6.2.2. Verwaltung der Benutzerpräferenzen
 - 6.2.3. Export von Präferenzen. Verwaltung von Genehmigungen
- 6.3. Datei-Speichersystem
 - 6.3.1. Klassifizierung von Dateisystemen auf mobilen Geräten
 - 6.3.2. Internes Dateisystem
 - 6.3.3. Externes Dateisystem
- 6.4. JSON-Dateien als Speicher in Android
 - 6.4.1. Unstrukturierte Informationen in JSON-Dateien
 - 6.4.2. Bibliotheken für die Verwaltung von Daten mit JSON
 - 6.4.3. Verwendung von JSON in Android. Empfehlungen und Optimierungen

- 6.5. XML-Dateien als Speicher in Android
 - 6.5.1. XML-Format in Android
 - 6.5.2. XML durch SAX-Bibliotheken
 - 6.5.3. XML durch DOM-Bibliotheken
 - 6.6. SQLite-Datenbanken
 - 6.6.1. Relationale Datenbank für die Datenverwaltung
 - 6.6.2. Nutzung der Datenbank
 - 6.6.3. SQLite-Methoden für die Datenverwaltung
 - 6.7. Fortgeschrittene Nutzung von SQLite-Datenbanken
 - 6.7.1. Wiederherstellung bei Fehlern mittels Transaktionen in SQLite
 - 6.7.2. Verwendung des Caches zur Beschleunigung des Datenzugriffs
 - 6.7.3. Datenbank für Mobilgeräte
 - 6.8. Bibliothek Room
 - 6.8.1. Architektur der Bibliothek Room
 - 6.8.2. Bibliothek Room. Funktionsweise
 - 6.8.3. Bibliothek Room: Vorteile und Nachteile
 - 6.9. *Content Provider* für die Weitergabe von Information
 - 6.9.1. *Content Provider* für die Weitergabe von Information
 - 6.9.2. *Content Provider* in Android. Techniker im Einsatz
 - 6.9.3. Sicherheit bei *Content Provider*
 - 6.10. Datenerfassung in der Internet-Cloud
 - 6.10.1. Android und Cloud-Speichersysteme
 - 6.10.2. SOAP und REST Dienste für Android
 - 6.10.3. Fragen zu verteilten Systemen
 - 6.10.4. Das Internet als Sicherheitskopie für Anwendungsdaten
- Modul 7. Tools für Android-Geräte**
- 7.1. Verwaltung: Tools "TO DO"
 - 7.1.1. Marktinstrumente
 - 7.1.2. Marktinstrumente. Vergleich der Funktionalitäten
 - 7.1.3. Management-Tools Unterschiede
 - 7.2. MDM: Verwaltung von Unternehmensgeräten
 - 7.2.1. Kontrolle von Unternehmensgeräten
 - 7.2.2. Analyse der wichtigsten Marktinstrumente
 - 7.2.3. Wahl der Instrumente
 - 7.3. CRM: Marktinstrumente
 - 7.3.1. Analyse von Marktinstrumenten mit Android-Anwendung
 - 7.3.2. Marktinstrumente. Effizienz
 - 7.3.3. Marktinstrumente. Verwendungen
 - 7.4. Drohnen mit Android
 - 7.4.1. Apps für Android-Geräte zur Steuerung von Drohnen
 - 7.4.2. Eigenständige Kontrollen
 - 7.4.3. Drohnenanwendungen auf Android
 - 7.5. Android, Mehrwert bei Bankplattformen
 - 7.5.1. Android in Bankplattformen
 - 7.5.2. Risiken und Betrug durch Cyber-Kriminelle
 - 7.5.3. Verwendung von mobilen Geräten
 - 7.6. *Brokering* auf mobilen Geräten
 - 7.6.1. Marktinstrumente und ihr Einsatz
 - 7.6.2. Vergleich der Werkzeuge
 - 7.6.3. Wahl des Werkzeugs für jeden Einsatz
 - 7.7. Unterhaltungs- und Schulungsinstrumente
 - 7.7.1. Verwendungen
 - 7.7.2. Marktinstrumente
 - 7.7.3. Funktionsvergleiche zwischen Android-Entwicklungstools
 - 7.8. IoT Android
 - 7.8.1. *Framework* und Marktplattformen
 - 7.8.2. Android IoT-Risiken und Überlegungen
 - 7.8.3. Anwendungen des IoT auf Android
 - 7.9. Effizienz des Prozesses
 - 7.9.1. Analyse der Marktinstrumente für die Erstellung von Apps
 - 7.9.2. Vergleich von Tools zur Erstellung von Android-Anwendungen
 - 7.9.3. *Use Case*

- 7.10. Die am häufigsten heruntergeladenen Anwendungen von heute
 - 7.10.1. Die am häufigsten heruntergeladenen Tools von heute
 - 7.10.2. Gruppierung nach Familien
 - 7.10.3. Hauptverwendung, sekundäre Verwendung und Vergleich mit IOS

Modul 8. Design Responsive auf Android

- 8.1. *Responsive Design*
 - 8.1.1. *Design Responsive*
 - 8.1.2. Benutzerfreundlichkeit, Zugänglichkeit und UX
 - 8.1.3. *Design Responsive*. Vor- und Nachteile
- 8.2. *Mobile vs. Tablet vs. Web vs. Smartwatches*
 - 8.2.1. Verschiedene Formate, verschiedene Größen, verschiedene Bedürfnisse
 - 8.2.2. Fragen der Gestaltung
 - 8.2.3. Adaptiv vs. Responsive
- 8.3. Stilrichtlinien
 - 8.3.1. Stilrichtlinien. Nützlichkeit
 - 8.3.2. *Material Design*
 - 8.3.3. Eigene Stilrichtlinien
- 8.4. *Flexibles Layouting*
 - 8.4.1. *Flexibles Layouting*
 - 8.4.2. Grundlegendes *Layouting*
 - 8.4.3. *Layouting in Grid*
 - 8.4.4. *Layouting mit Relative Layout*
 - 8.4.5. *Layouting mit Constraint Layout*
- 8.5. Flexible Hilfsmittel
 - 8.5.1. Flexible Hilfsmittel
 - 8.5.2. Bilder
 - 8.5.3. 9-Patch
 - 8.5.4. Globale Ressourcen
- 8.6. Flexible Navigation
 - 8.6.1. Flexible Navigation
 - 8.6.2. Navigation mit *Aktivitäten*
 - 8.6.3. Navigation mit *Fragments*



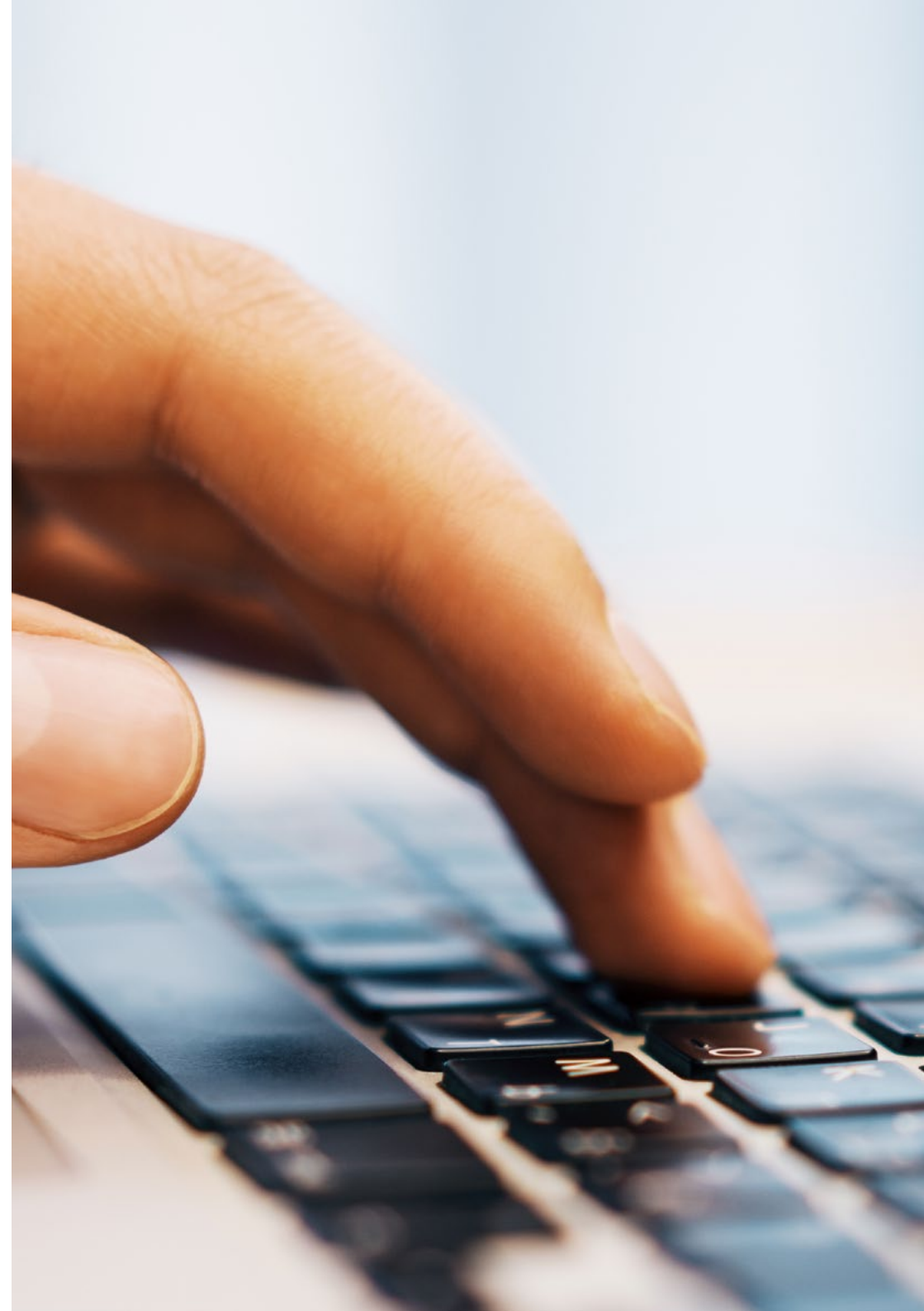


- 8.7. Externe Tools
 - 8.7.1. Automatische Generatoren
 - 8.7.2. Prototyping-Tools
 - 8.7.3. Design Tools
- 8.8. Debug und Tests
 - 8.8.1. *Debug Layouts*
 - 8.8.2. Automatische Tests
 - 8.8.3. Komponentenbasierte Entwicklung
 - 8.8.4. Gute Praktiken in *Testing und Prüfungen*
- 8.9. Alternativen zum nativen Android I. Web-Seiten
 - 8.9.1. Design in einer WebView
 - 8.9.2. *Chrome Custom Tabs*
 - 8.9.3. *Debug und Tests auf Web Seiten*
- 8.10. Alternativen zum nativen Android II. Hybride Anwendungen
 - 8.10.1. *React/React native*
 - 8.10.2. *Flutter*
 - 8.10.3. *Ionic*
 - 8.10.4. Apache Cordova

Modul 9. Marketing in Android-Anwendungen

- 9.1. Vom *Customer Service* bis *Customer Experience*
 - 9.1.1. *Customer Service*. Entwicklung des Kunden von heute
 - 9.1.2. Benutzer mit Zugang zu Informationen. Anforderungen und Bedürfnisse
 - 9.1.3. Feedback als Wissensquelle
- 9.2. *Customer Journey*
 - 9.2.1. Benutzerpfad zur Konvertierung
 - 9.2.2. Mikro-Segmentierung
 - 9.2.3. Kanalübergreifende Erfahrung
- 9.3. Messung der Benutzererfahrung
 - 9.3.1. Web- und mobile Architektur
 - 9.3.2. Sitzungsanalyse als neuer Standard
 - 9.3.3. Stand der Technik Benutzererfahrung

- 9.4. Marketing für Android-Anwendungen
 - 9.4.1. CX+IA
 - 9.4.2. CX+Blockchain
 - 9.4.3. CX+IoT
- 9.5. CX-Produkte (Kundenerfahrung)
 - 9.5.1. Standards der Industrie
 - 9.5.2. Telepräsenz
 - 9.5.3. Kundenerfahrung für alle Entwicklungsagenten
- 9.6. Benutzerzentrierte Arbeit
 - 9.6.1. Geräte
 - 9.6.2. Design Thinking
 - 9.6.3. Arbeit vor Ort
- 9.7. Benutzerwissenschaft
 - 9.7.1. Benutzerwissenschaft. Goldene Regeln
 - 9.7.2. Iteration
 - 9.7.3. Häufige Fehler
- 9.8. Prototyping und Wireframing
 - 9.8.1. Prototyping und *Wireframing*
 - 9.8.2. *Hands-On*
 - 9.8.3. Fortgeschrittenes Niveau
- 9.9. Mobile Benutzeroberflächen
 - 9.9.1. Visuelle Gestaltung. Regeln
 - 9.9.2. Benutzeroberflächen von Apps. Schlüssel
 - 9.9.3. Bewährte Praktiken bei der Entwicklung mobiler Schnittstellen
- 9.10. Bewährte Praktiken für die Benutzererfahrung. Tipps für Entwickler
 - 9.10.1. Stufe Eins. Bewährte Praktiken in CX
 - 9.10.2. Stufe Zwei: Bewährte UX-Praktiken
 - 9.10.3. Stufe Drei: Bewährte Praktiken für die Benutzeroberfläche



Modul 10. Lebenszyklus auf Android-Anwendungen. Cloud, Playstore und Versionierung

- 10.1. Lebenszyklus der Software
 - 10.1.1. Lebenszyklus der Software
 - 10.1.2. Agile Methodologien
 - 10.1.3. Der agile kontinuierliche Software Zyklus
- 10.2. Manuelle Produktentwicklung
 - 10.2.1. Manuelle Integration
 - 10.2.2. Manuelle Lieferung
 - 10.2.3. Manueller Einsatz
- 10.3. Überwachte Integration
 - 10.3.1. Kontinuierliche Integration
 - 10.3.2. Manuelle Überprüfung
 - 10.3.3. Automatische statische Überprüfungen
- 10.4. Logische Tests
 - 10.4.1. Einheitstests
 - 10.4.2. Integrationstests
 - 10.4.3. Leistungstests
- 10.5. Kontinuierliche Integration
 - 10.5.1. Kontinuierlicher Integrationszyklus
 - 10.5.2. Abhängigkeiten zwischen Integrationen
 - 10.5.3. Kontinuierliche Integration als Methode zur Verwaltung von Repositories
- 10.6. Kontinuierliche Lieferung
 - 10.6.1. Kontinuierliche Lieferung. Typologie der zu lösenden Probleme
 - 10.6.2. Kontinuierliche Lieferung. Lösung von Problemen
 - 10.6.3. Vorteile der kontinuierlichen Bereitstellung
- 10.7. Kontinuierlicher Einsatz
 - 10.7.1. Kontinuierlicher Einsatz. Typologie der zu lösenden Probleme
 - 10.7.2. Kontinuierlicher Einsatz. Lösung von Problemen

- 10.8. *Firebase Test Lab*
 - 10.8.1. Konfiguration von *GCloud*
 - 10.8.2. Konfiguration von Jenkins
 - 10.8.3. Verwendung von Jenkins. Vorteile
- 10.9. Konfiguration von *Gradle*
 - 10.9.1. Automatisierungssystem *Gradle*
 - 10.9.2. *Gradle Build Flavors*-Komponente
 - 10.9.3. *Gradle Lint*-Komponente
- 10.10. Lebenszyklus von Android-Anwendungen. Beispiel
 - 10.10.1. Konfiguration von *SemaphoreCI* und GitHub
 - 10.10.2. Konfiguration der Arbeitsblöcke
 - 10.10.3. Promotionen und *Deployment*



Warten Sie nicht länger mit der Professionalisierung, Sie sind ein Teil der Zukunft. Beginnen Sie mit der Entwicklung der besten mobilen Anwendungen mit diesem privaten Masterstudiengang

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 Anwendungsentwicklung für Android 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"*

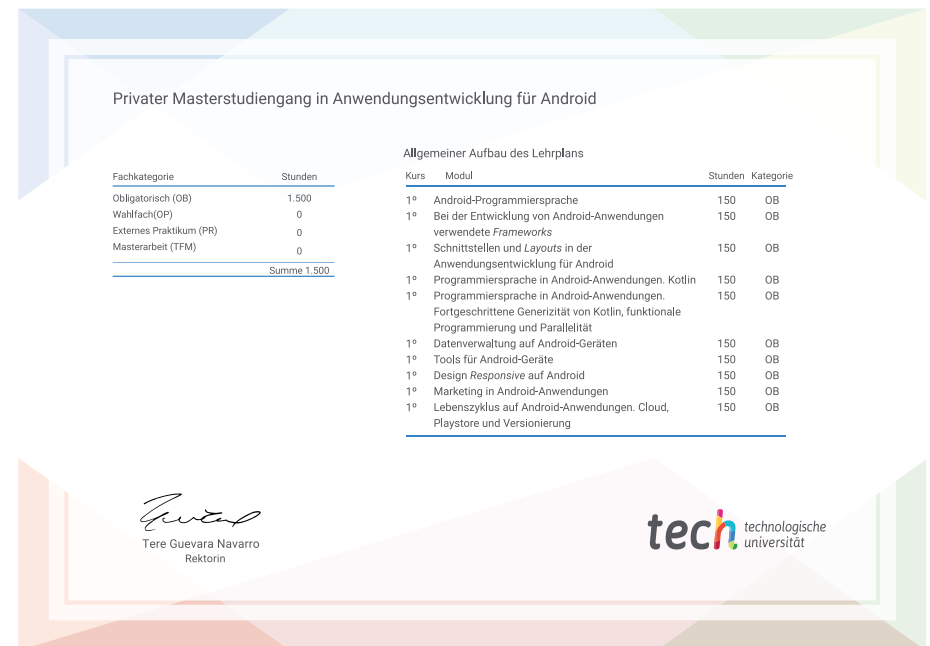
Dieser **Privater Masterstudiengang in Anwendungsentwicklung für Android** enthält das vollständigste und aktuellste Programm auf dem Markt.

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 Anwendungsentwicklung für Android**

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 institut
virtuelles Klassenzimmer sprachen

tech technologische
universität

Privater Masterstudiengang Anwendungsentwicklung für Android

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technologische Universität
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Privater Masterstudiengang Anwendungsentwicklung für Android

```
        e.stopPropagation();  
        e.preventDefault();  
        e.preventDefault();  
        e.preventDefault();  
        e.preventDefault();  
        return;  
  
        //up/down  
        if (e.keyCode == 38 || e.keyCode == 40) {  
            e.preventDefault();  
            e.stopPropagation();  
            if (e.keyCode == 38) { // up  
                // show previous search query  
                if (hist.currentIndex == hist.history.length) {  
                    hist.currentQuery = input.value;  
                    // skip previous search if we're already there  
                    if (hist.currentQuery == hist.history[hist.history.length - 1])  
                        hist.currentIndex--;  
                }  
            }  
        }  
    }  
    if (hist.currentQueryToday
```