Advanced Master's Degree

Mobile Application Development, Android Expert

tech technological university

tech technological university

Advanced Master's Degree

Mobile Application Development, Android Expert

- Modality: **online**
- Duration: **2 years**
- Certificate: **TECH Technological University**
- Dedication: **16h/week**
- Schedule: **at your own pace**
- Exams: **online**

Website: www.techtitute.com/us/information-technology/advanced-master-degree/advanced-master-degree-mobile-application-development-android-expert

Index

01

Introduction

Today it is impossible to imagine a world without apps and companies that take advantage of new technologies. Mobile applications are a step forward for any company, constituting a sign of maturity and an effort to adapt to new user habits, making them essential in almost any field. Currently, the need for mobile applications is linked to the functionalities they can offer beyond the information that a web page may have. That is why, today, there are more than 3 billion Android devices in the world and, consequently, the mobile application development sector has experienced a dramatic growth. Now than ever the industry is in dire need of professionals specialized in Application Development, an issue that has driven the creation of this program.

Learn from experts about all the key aspects of customer experience to develop Android applications with a strong market entry"

Labor market reports confirm the growing demand for expert profiles in the design of mobile applications, valuing the complete experience in the life cycle of development, deployment and monetization of applications. To equip the computer scientist in all knowledge about application programming language, with special focus on Android, architectures and user interfaces. This program has been designed by the best experts in the field, who make up the teaching staff.

Thus, the computer scientist will acquire the skills to understand the different programming languages for each type of device, delving into *responsive* design. This will allow you to develop applications adaptable to the different characteristics of each device. In turn, the professional will delve into areas such as web computing, software, business development, marketing and sales, consulting, multi-platform mobile technology, business, automotive, home automation, Internet of Things (IoT), banking and drones.

In addition, TECH has the best 100% online study methodology, which eliminates the need to attend classes in person or follow a predetermined schedule. In this way, in 24 months the computer scientists will acquire the precise basis to develop their own business from the development of applications or to undertake consulting tasks in all aspects related to Android mobile technology.

This **Advanced Master's Degree in Mobile Application Development, Android Expert** contains the most complete and up-to-date program on the market. The most important features include:

- The development of case studies presented by IT experts
- The graphic, schematic, and practical contents with which they are created, provide scientific and practical information on the disciplines that are essential for professional practice
- Practical exercises where self-assessment can be used to improve learning
- Special emphasis on innovative methodologies in the development of Mobile Applications
- Theoretical lessons, questions to the expert, debate forums on controversial topics, and individual reflection assignments
- Content that is accessible from any fixed or portable device with an Internet connection

In this Advanced Master's Degree you will be up to date and know in depth the different programming languages for each type of mobile device"

As you master all the content on mobile app development, you will understand how to monetize them and delve deeper into mobile marketing"

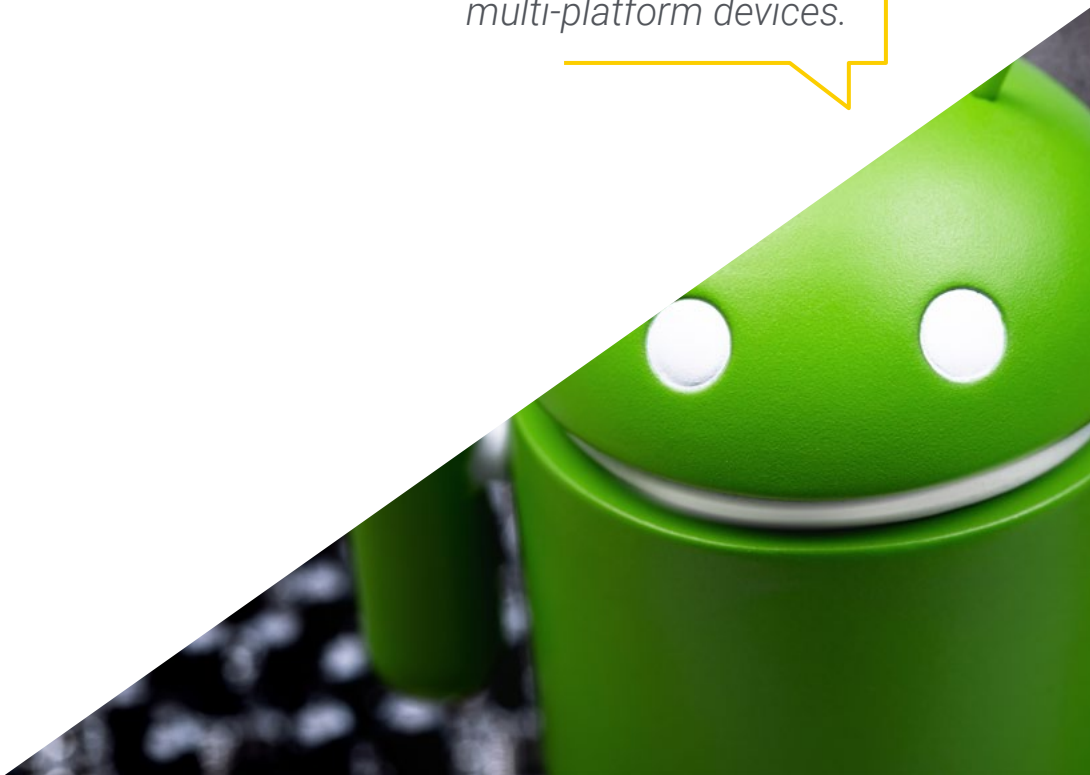
Provide efficient solutions to your company, handling the fundamentals related to Android Application Development Systems.

TECH gives you the opportunity to learn how to develop mobile applications in an autonomous and professional way, in multi-platform devices.

Its teaching staff includes professionals from the field of information technology, who bring to this program the experience of their work, as well as recognized specialists from leading companies and prestigious universities.

The multimedia content, developed with the latest educational technology, will provide the professional with situated and contextual learning, i.e., a simulated environment that will provide an immersive learning experience designed to prepare for real-life situations.

This program is designed around Problem-Based Learning, whereby the student must try to solve the different professional practice situations that arise throughout the program. For this purpose, the professional will be assisted by an innovative interactive video system created by renowned and experienced experts.



02

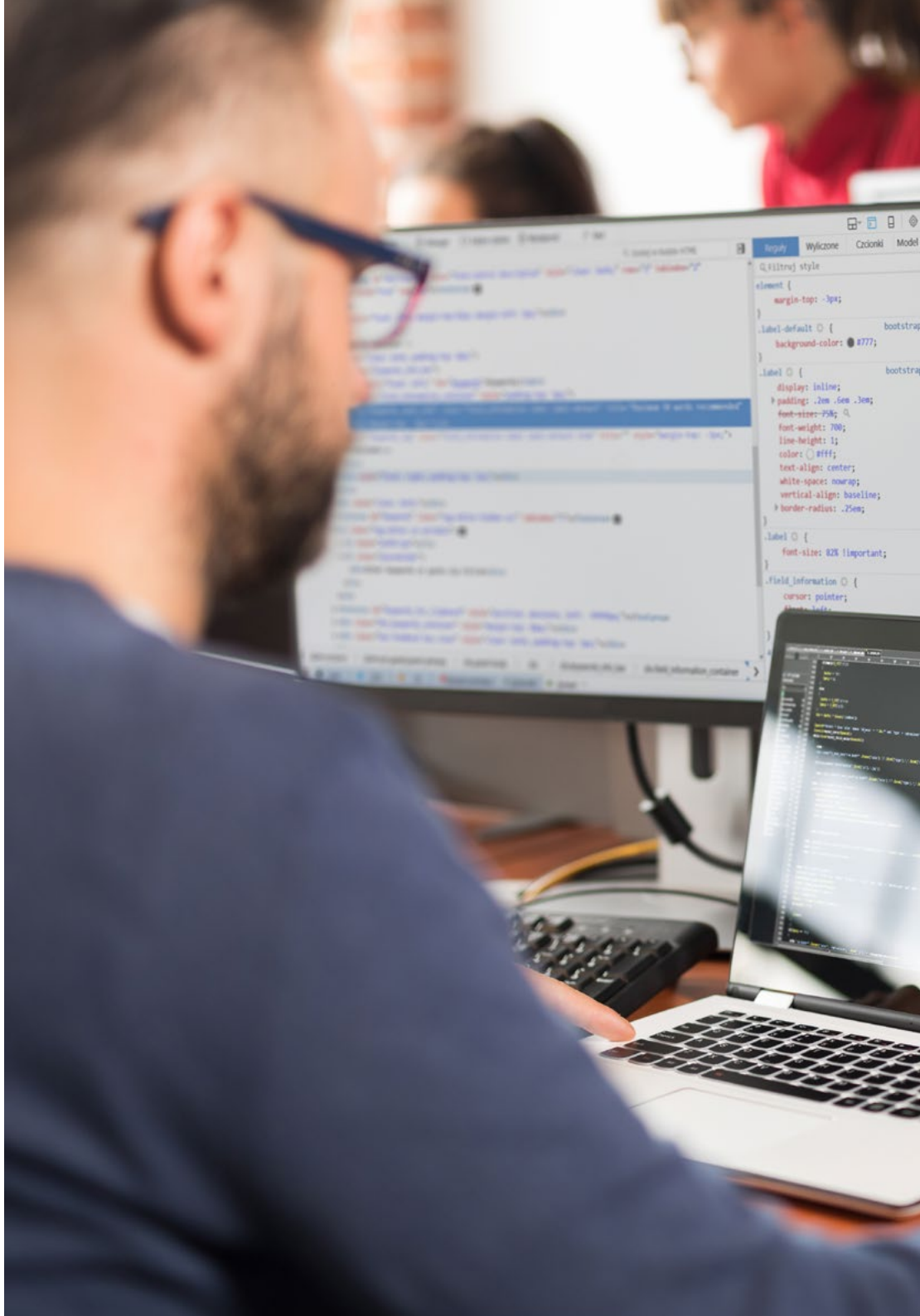
Objectives

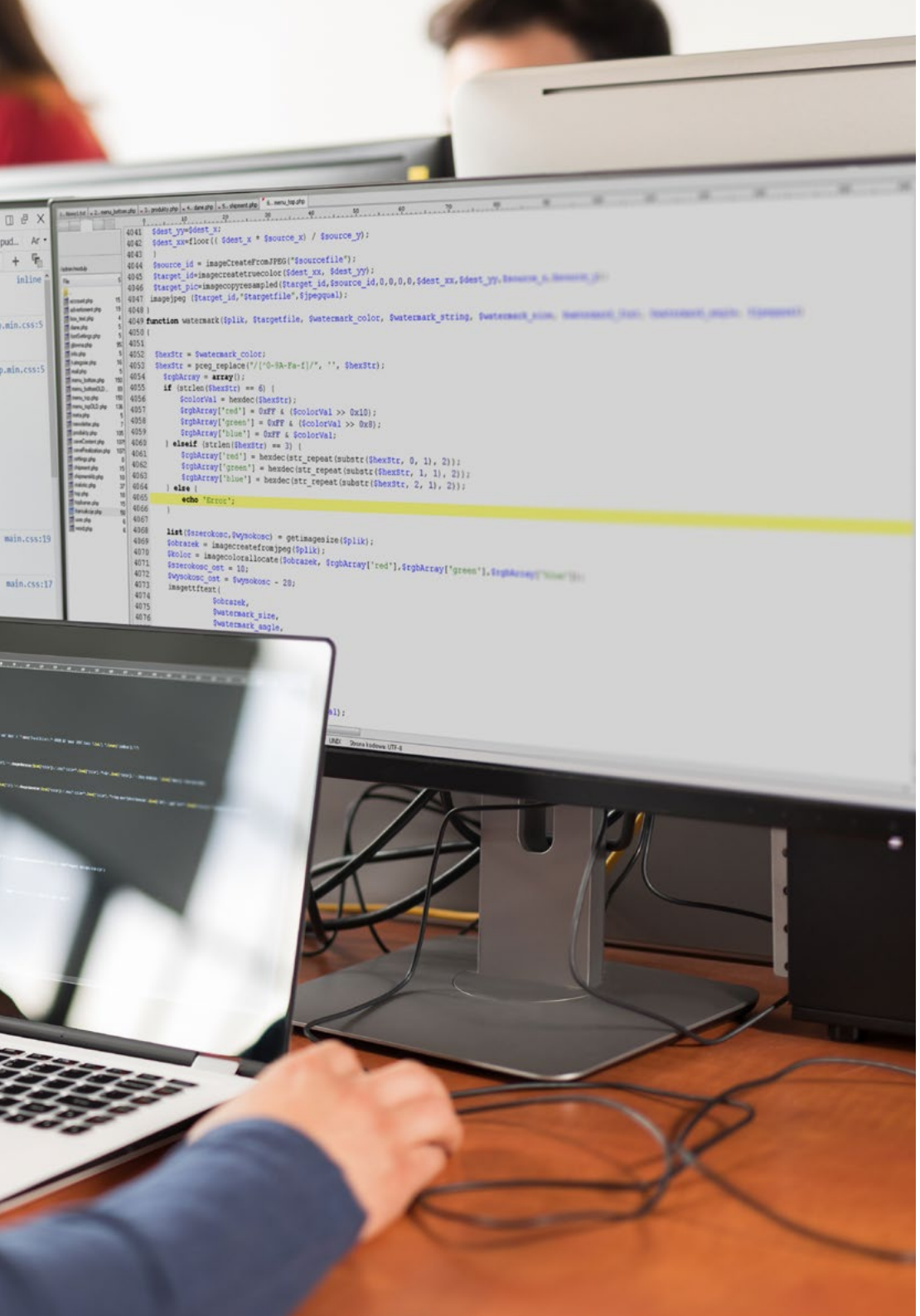
Due to the notorious growth and consolidation that the world of mobile technologies and devices brings with it, the computer scientist is obliged to continuously update their skills in order to acquire all the necessary knowledge. Therefore, this program provides you with all the necessary tools and fundamentals to deepen in the development of mobile applications, with special focus on the Android system, and provide effective solutions in the professional environment. Graduates with this qualification will be up to date with the latest trends and will know how to perform in all aspects from design and programming to the final user experience.

Don't miss the opportunity to update your knowledge with this Advanced Master's Degree and be able to provide effective solutions in the professional environment"

General Objectives

- Analyze user needs and behavior in relation to mobile devices and their applications
- Execute the design of architectures, iterations and user interfaces through the programming languages of the most representative mobile platforms on the market (Web, iOS and Android)
- Apply error control, testing and debugging mechanisms in mobile application development
- Address different practical and business cases for publishing, distributing and disseminating mobile applications in the main application markets
- Master the practical knowledge to plan and manage technology projects related to mobile technologies
- Develop the skills, aptitudes and tools necessary to learn to develop mobile applications in an autonomous and professional manner, on multi-platform devices





- Explore content related to app monetization and mobile marketing
- Determine the structural elements of an Android system
- Analyze the differences between the different development *frameworks*, strengths and weaknesses
- Develop advanced capabilities and best practices in Application Development in the Kotlin Programming Language
- Develop a methodology for optimal data management on the device
- Analyze use cases for Android devices on the market
- Master the elements of responsive design and overcoming the challenges associated with it
- Compile the different stages of a continuous integration cycle focused on Android development

Specific Objectives

Module 1. Programming Methodologies in Mobile Application Development

- Explore traditional software development processes
- Analyze agile development processes
- Promote development practices
- Examine the different representation and diagramming techniques
- Deepen in the different design patterns present in the software industry
- Explore different software testing techniques
- Recognize the rules and standards of quality reference in development

Module 2. Technologies in Mobile Application Development

- Establish concepts for mobile devices
- Compile the main platforms
- Examine their common components
- Identify differentiating components, their capabilities and limitations
- Define the different scenarios in which they can operate Advantages
- Analyze the different interactions that these devices can mediate
- Raise awareness of the different abuses that can be committed

Module 3. Work Tools for Mobile Application Development

- Prepare the development environment
- Acquiring command terminal skills
- Efficient use of the version control system
- Address the use of remote code versioning systems
- Establish the key notions of Internet operation
- Develop relevant software programming concepts
- Examine data structures
- Review algorithm design and interpretation techniques

Module 4. Multi-Platform Web Development for Mobiles

- Determine the advantages and limitations of the native and hybrid App development model
- Examine the features and limitations of Progressive Web Apps (PWA)
- *Analyze the main Frameworks* for web application development: Angular, React, Vue
- Compile the main technologies for the development of multi-platform mobile applications *Ionic* and *Flutter*
- Analyze capabilities to deploy these hybrid apps as Web or Desktop Apps on PCs
- Examine a model to choose the alternative best suited for the development of a specific application

Module 5. Databases for Mobile Application Development

- Identify the best database model in relation to the characteristics of the mobile application
- Establish the capabilities of each of the database systems
- Determine the differences between the different databases
- Examine how to connect to and load/extract data from different types of databases
- Analyze the basic capabilities of development environments with database capabilities including

Module 6. Application Development for iOS Systems

- Develop an application in Swift
- Use Cocoa Pods to manage libraries
- Make use of Alamofire to connect our application with a RESTful API
- Specify the basic requirements for the choice of a library
- Develop a monetization system with ADMOB
- Design views from code
- Publish an application in the App Store

Module 7. Continuous Integration Deployments for Mobiles

- Determine the worst case scenario that gives rise to the need for this methodology
- Specify the requirements that the software must meet to be integrated
- Establish what is continuous integration, continuous delivery and continuous deployment
- Analyze DevSecOps
- Examine continuous monitoring
- Develop the implementations of the different stages

Module 8. Mobile *User Experience*

- Analyze the new type of user, their interactions and their journey through mobile applications and websites
- Determine the fundamental tools for web analytics, mobility and accessibility
- Specify micro-interaction assessment techniques and the design of customized experiences
- Establish how new disruptive technologies such as AI or IoT have taken customer experience to new standards
- Show how behavioral analytics generates a quantity and quality of data never seen in traditional analytics
- Develop new methodologies such as *Design Thinking*,, focused on the user
- Propose basic and advanced prototyping and *wireframing* tools

Module 9. Security on Mobile Devices

- Determine the security features and levels of security on a mobile device
- Establish the techniques to be used for a device
- Analyze the common errors in security
- Examine the mechanisms enabled in programming to avoid security breaches
- Specify the recommendations given by the Security Agencies
- Compile the different solutions available on the market for managing the security of mobile devices within the enterprise
- Analyze the cryptographic processes applied to mobile security

Module 10. Android Programming Language

- Examine the Linux kernel and virtual machine on the Android base
- Analyze native system libraries
- Establish the benefits of Android over other platforms
- Determining the elements of an Android application
- Introduce Android versions and their enhancements
- Evaluate the market for Android applications
- Fundamentals of Android's future evolution

Module 11. *Frameworks* Used in Android Application Development

- Analyze the Android Core *framework*
- Develop other *frameworks* used for Android application development.
- Implement libraries with Gradle
- Set up the *frameworks* to connect to an API
- Generate specialized knowledge on Architectures as MVP
- Clarify the pros and cons of MVP and MVVM

Module 12. Interfaces and *Layouts* in Android Application Development

- Introduce the view life cycle model in Android
- Examining the most important attributes of a visual design (*layout*)
- Analyze available *layout* designs
- Create a reusable *layout* design
- Determine how to use alternative resources
- Identify the differences in the use of these components compared to other programming systems
- Establish the potential and use of the *AndroidManifest.xml* file

Module 13. Programming Language in Android Applications Kotlin

- Develop the Kotlin programming language
- Compile the features and differential capabilities of the Kotlin language
- Examine the basic execution model of a Kotlin program
- Analyze the language syntax and program structure of a Kotlin program
- Specify the model of types and variables in Kotlin
- Establish the various forms of code flow management in Kotlin
- Determine the model of classes, collections and objects in Kotlin
- Generate specialized knowledge about the inheritance model in Kotlin
- Specify the exception and null type management model in Kotlin

Module 14. Programming Language in Android Applications Advanced Kotlin Genericity, Functional Programming and Parallelism

- Examine the covariant and contravariant genericity model in Kotlin
- Analyze Kotlin's functional programming model with *Lambdas*
- Define Kotlin's higher order functions
- Develop extensions and companion objects in Kotlin
- Examine the use of the *delegation* pattern in Kotlin
- Compile annotations and reflection in Kotlin
- Deepen the *testing* model in Kotlin
- Establish the different asynchronous programming models available in Kotlin
- Determine the Kotlin coroutines model
- Compile the various libraries and utility tools of the Kotlin ecosystem

Module 15. Data Management on Android Devices

- Analyze the different techniques for data management on Android
- Propose methods for optimal use of data on the device
- Identify the tools required for data optimization
- Examine the features of JSON and XML for Android data management
- Evaluate general distributed systems issues applicable to the world of mobile device applications
- Determine the use of the *room* library as an abstraction for the use of SQLite on Android and its advantages and disadvantages
- Establish the necessary security permissions in data management in any of the techniques used in Android

Module 16. Android Device Tools

- Compile the most advanced tools in day-to-day management
- Evaluate Android device control tools
- Demonstrate the advantages of using Android on drones
- Specify the functionalities of CRM tools
- Demonstrate the benefits of Android devices in banking platforms
- Analyze the possibilities of IoT and Android platforms
- Examine process efficiency tools in Android

Module 17. *Responsive*Design in Android

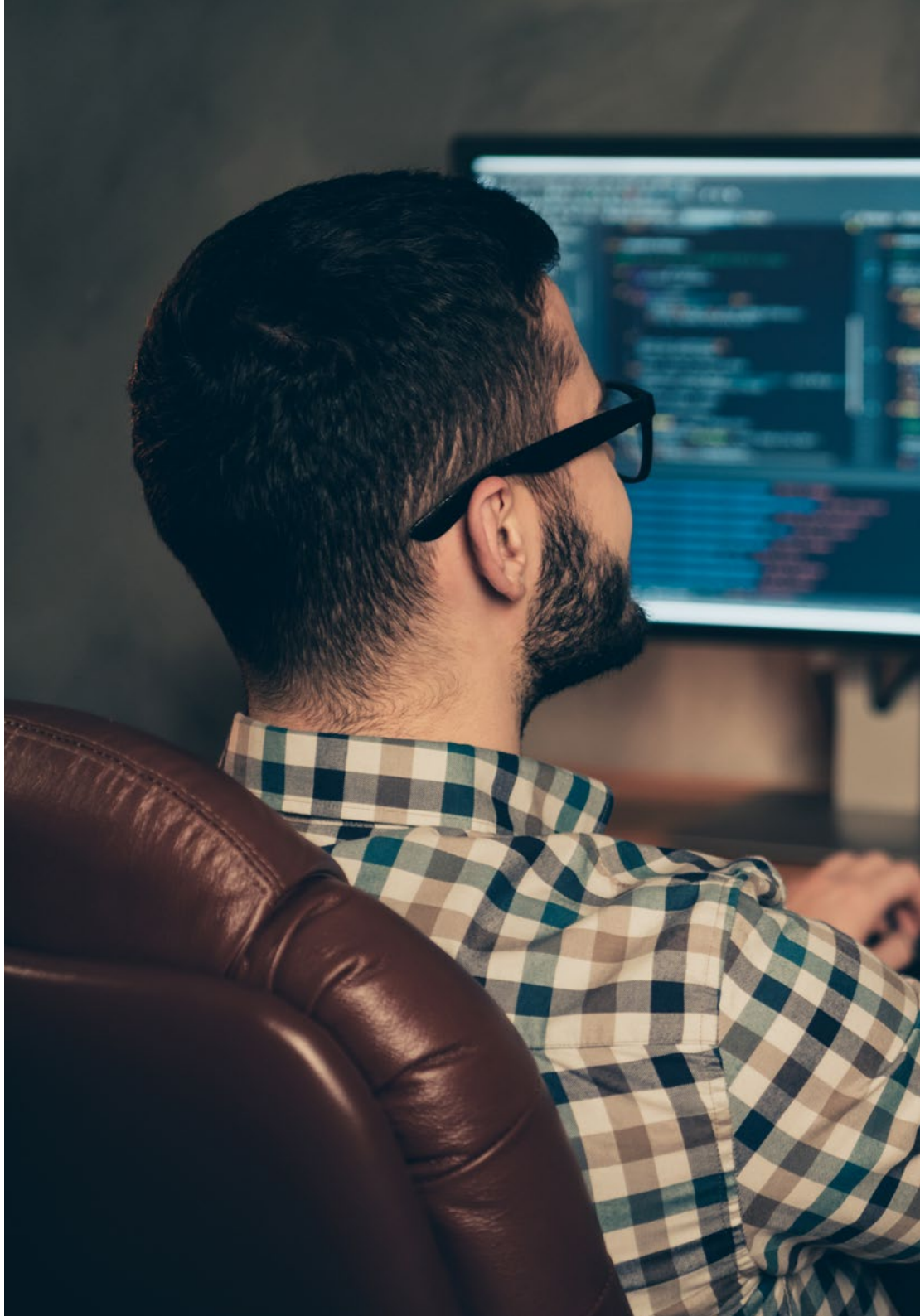
- Analyze the main elements of a design
- Define a visual design methodology and a screen design methodology
- Solve the various problems encountered in multi-device development
- Have tools to generate better and more resources for multi-device development
- Examine alternative *frameworks* to native responsive development
- Generate a unique methodology to develop applications using best practices for multi-device visualization from the beginning of the project

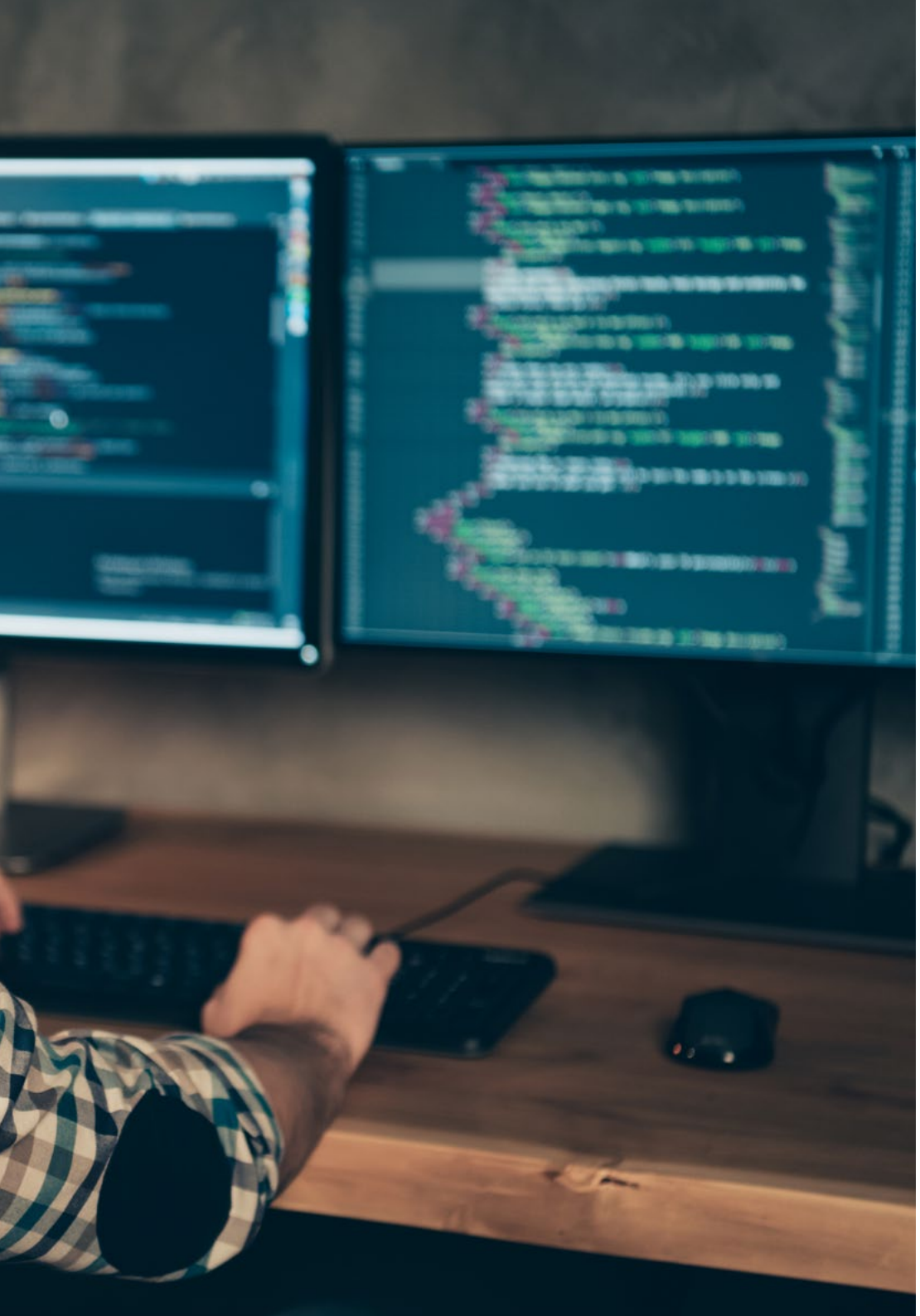
Module 18. Android Application Marketing

- Analyze new user-centric methodologies
- Determine how Artificial Intelligence has taken CX to the next level
- Establish the importance of accessibility and mobility
- Develop state-of-the-art session and behavioral analytics techniques
- Specify micro-personalization objectives during the user *journey*
- Compile new methodologies for a changing and lively environment
- Propose prototyping techniques

Module 19. Android Application Life Cycle: Cloud, Playstore and Versioning

- Realize the benefits of adopting an automated release deployment model
- Establish the differences between continuous integration, continuous delivery and continuous deployment
- Define the main features of DevOps
- Assess some of the fundamental tools for implementing CI/CD pipelines
- Identify the essential factors for developing applications ready to support CI/CD processes
- Examine Container Technologies as a fundamental pillar of CI/CD
- Identify practices, use cases, technologies and tools of the CI/CD ecosystem





You will acquire specialized knowledge about the Android application lifecycle in order to work more efficiently as a developer"

03

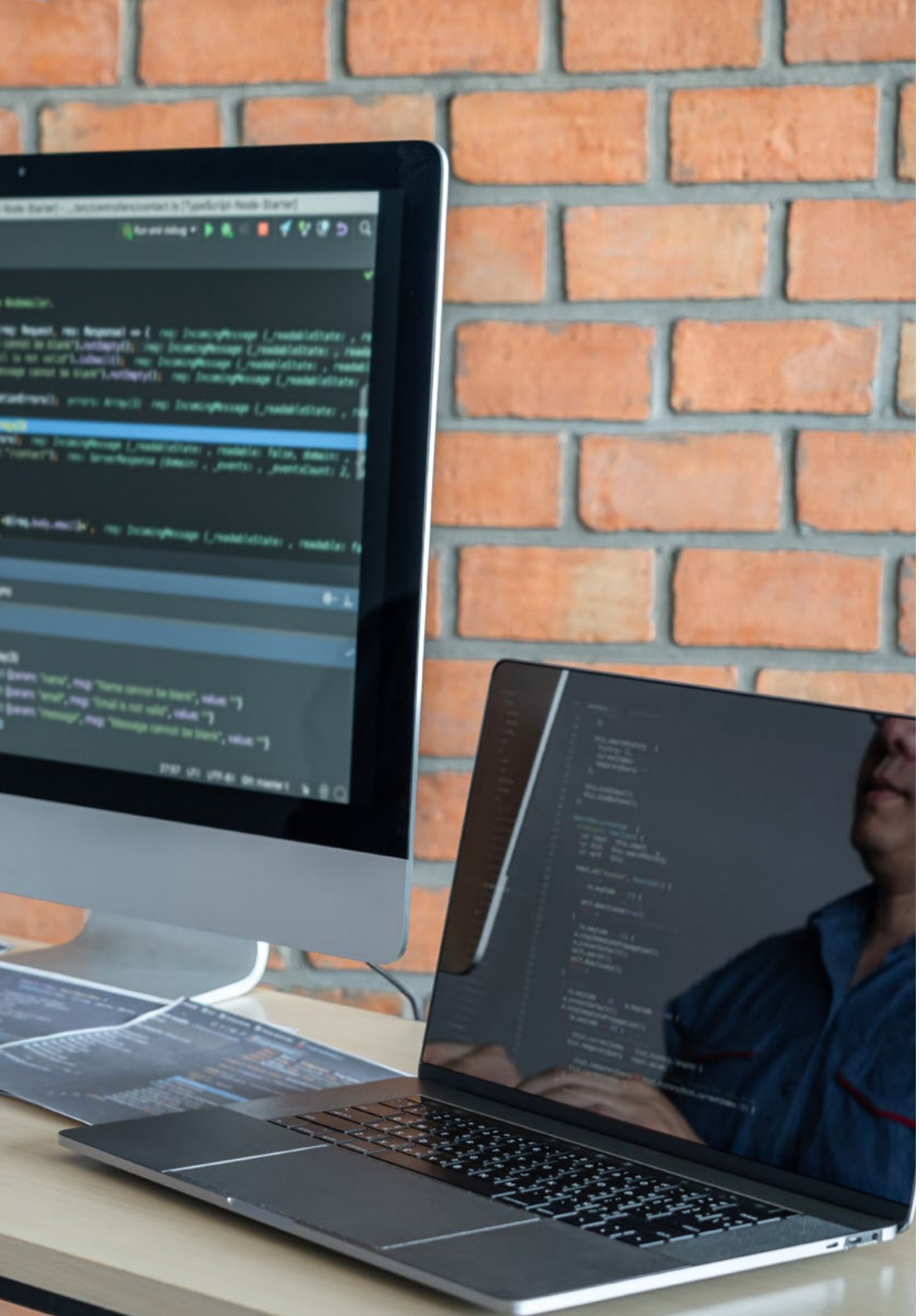
Skills

The technology industry constantly demands multitasking professional profiles, which implies constant training to perform in a market with a vertiginous evolution. This Advanced Master's Degree trains computer scientists with specialized knowledge, updated on the different technology components of mobile devices to develop applications. The graduate will have an in-depth knowledge of the best *hardwares* annd *softwares* essential for any professional in this sector In this way, the computer scientist will be up to date in the knowledge necessary to be autonomous when developing an application for Android devices from design to production.

With this qualification you will boost your career by gaining the practical knowledge to plan and manage projects with mobile technologies"

General Skills

- Develop mobile applications for both Android and iOS mobile devices
- Enhance the necessary *skills* at all points of the software development life cycle
- Analyze the different programming methodologies for different devices and use cases
- Master specialized knowledge of the different technology components of mobile devices
- Understand the working environments for different mobile application programming languages and styles
- Efficient realization of applications from a user experience point of view to make it truly functional and appealing
- Master the structural elements of an Android system
- Analyze the various *frameworks* to be used by multiple Architectures
- Define the properties of an interface from a usability point of view
- Develop specialized knowledge about the Kotlin programming language and the context of its emergence
- Manage the *content provider* for data exchange and data security
- Address enterprise tools on Android: their efficiency, usefulness
- Analyze the elements of a responsive design
- Explore new user-centered methodologies
- Adjusting the Automated Pipeline System to the Android Ecosystem



Specific Skills

- Analyze Software Development Processes from the traditional and agile point of view
- Develop general considerations for mobile devices
- Master the key concepts of programming and the Internet, the Web and how it works
- Compare the native development model and the cross-platform web development model based on hybrid apps
- Determine how to use databases in mobile applications
- Publish an application in *Play Store*
- Determining the different stages of a continuous integration cycle
- Understand the principles of user-centric culture and how it creates a new position for customer experience professionals
- Address security issues on mobile devices
- Utilize alternative resources by analyzing available *layout* designs
- Develop the various forms of code flow management in Kotlin and their capabilities
- Improve application productivity through the differential capabilities of the Klotin language
- Develop extensions and companion objects in Kotlin
- Use databases or network services to create files in different formats
- Handling the *room* library as an abstraction for using SQLite in Android
- Evaluate Android device control tools, analyzing IoT and Android platform possibilities
- Adopt *responsive* design to provide more satisfying user experiences
- Combine Prototyping and *Wireframing* techniques with new disruptive technologies such as Artificial Intelligence and the Internet of Things

04

Course Management

This Advanced Master's Degree in Mobile Application Development, Android Expert is led by teachers with extensive knowledge and experience in new technologies, solution architecture and digital infrastructure, experts in Android Programming and Application Developers. In this way, they offer the computer scientist a guarantee of the quality of the content selected for this qualification. Betting on the optimization of the learning process of professionals who are looking for the contribution they need for their professional success.

Compiling all the experiences of computer scientists and engineers who are experts in Application Development will add value to your proposals and make you more attractive to the job market"

International Guest Director

Colin Lee is a **successful mobile application developer**, specializing in **native Android code**, whose influence extends internationally. The Postgraduate Diploma is an **authority in the Twin Cities area and in the handling of Kotlin**. One of his most recent contributions was to demonstrate, in live code, how to **quickly build a browser** using the aforementioned programming language and Mozilla's open source browser components for Android.

In addition, his applications have been linked to globally significant companies. For example, he was in charge of **creating digital solutions for Pearson**, one of the largest international publishers. He also developed a low-level Android video recorder for the startup Flipgrid, later acquired by Microsoft.

He also built a successful Android VPN for a large **client in the consulting world**. In turn, he is the creator of a freight management tool implemented by the transnational **Amazon** to facilitate the work of its contracted truckers. On the other hand, he has helped build the **mobile versions of the Firefox browser** for Mozilla.

Today, he performs work as a contractor, including **code reviews and security checks**. His impact on mobile application development and his experience over the years make him a leading figure in the global technology arena.

Mr. Lee, Colin

- Director at ColinTheShots LLC
- Android Software Engineer for Specto Inc.
- Senior Android Engineer for Mozilla
- Software Development Engineer for Amazon
- Mobile Application Engineer for Flipgrid
- Software Configuration Specialist for Pearson VUE
- Bachelor's Degree from the University of Florida

Thanks to TECH you will be able to learn with the best professionals in the world"

Management

D. Olalla Bonal, Martín

- *Blockchain Technical Specialist* at IBM SPGI
- Digital Electronics Technician
- *Blockchain* Architect
- Infrastructure Architect in Banking
- *Hyperledger Fabric* training to companies
- Business-oriented companies *Blockchain* training
- Project management and implementation of solutions
- More than 25 years of experience in the IT world

Professors

Mr. Gozalo Fernández, Juan Luis

- Computer Engineer
- Associate Professor in DevOps University Experts and *Blockchain* Application Development at the International University of La Rioja.
- Former Blockchain DevOps Director at Alastria
- Director of Service Level Technology at Santander Spain
- Tinkerlink Mobile Application Development Manager at Cronos Telecom
- IT Service Management Technology Director at Barclays Bank Spain
- Bachelor's Degree in Computer Engineering from UNED
- Deep Learning Specialization in DeepLearning.ai

D. Gómez Rodríguez, Antonio

- Cloud Solutions Engineer at Oracle
- Project Manager in important business groups such as Sopra Group and Everis
- Project Manager at "Empresa pública de Gestion de Programas Culturales" at the Council of Culture of Andalusia
- Degree in Telecommunications Engineering from the Polytechnic University of Catalonia.
- Postgraduate Degree in Information Technologies and Systems, Catalan Institute of Technology
- E-Business Master, La Salle School of Business
- Information Systems Analyst. Sopra Group

Mr. Guerrero Díaz-Pintado, Arturo

- R&D Network Engineer at Telefónica
- Technical Presales Engineer across *Watson Customer Engagement* portfolio (Marketing and *Customer Experience* solutions) within Spain, Portugal, Greece and Israel at IBM.
- Professional services consultant working with leading organizations in Europe, Middle East and Latin America since IBM.
- Degree in Telecommunications Engineering from the University of Alcalá and the *Danish Technical University*
- He has made outstanding collaborations in renowned universities and higher education centers in subjects related to technology such as Artificial Intelligence, *Internet of Things, Cloud*, *Customer Experience y Digital Transformation*.

Mr. Villot Guisán, Pablo

- Cloud Architect, Exponential Solutions and Subject Mater Expert *Blockchain* at KPMG
- Cloud Architect, Exponential Solutions and Subject Mater Expert *Blockchain* Integration at Everis
- Developer and Technical Manager of web and heavy desktop applications for the Commercial Logistics area of Inditex, Connectis.
- Microsoft MSCA certification: *Cloud Platform*
- Degree in Computer Engineering from the University of La Coruña

Mr. Pérez Rico, Javier

- Android Technical Lead at Nologis
- Android Technical Lead at Seekle
- Androif programmer at Gowex-Ideup
- Junior Android Programmer at Tecnocom
- Speaker at the II iTest Symposium, E@tic2011
- Degree in Technical Computer Engineering of Systems at the Complutense University of Madrid
- Master's Degree in Research at the Complutense University of Madrid

Mr. Noguera Rodríguez, Pablo

- Expert Java Developer: JSE, JEE and Android - Ilabora Formación
- Android Applications Programming - EOI - Madrid
- Native App Developer (iOS & Android)- Starman Aviation (Aviaze App)
- Native App Developer (iOS) - Stef (Mtrack App)
- Native App Developer (iOS & Android) - Bitnovo (Bitnovo App)

Mr. Gutiérrez Blanco, Víctor

- Actual iOS Team Leader at DriveSmart
- iOS Team Leader at Keytree Spain
- iOS Leader at SeekleLabs
- iOS & Python Developer at Web Partners
- Full-Stack Developer at Let's Health
- Front-End Lead Developer at Indra and at Intelygenz
- Degree in Computer Engineering from the Pontificia University of Salamanca Campus Madrid

Mr. Frias Favero, Pedro Luis

- CTO - Swearit technologies
- COO - Key identification
- Full Stack Developer - Ironhack
- Bachelor's degree in Industrial Engineering from Yacambu University
- Expert in *Blockchain* and decentralized applications used by Alcalá University

Mr. Jiménez Pérez, Carlos

- Senior Android Developer at OnTheSpot - Telefónica Tech
- Associate Professor at Carlos III University of Madrid.
- Degree in Automatic and Electronic Engineering.
- Master's Degree in Electronic Systems and Applications Engineering

05

Structure and Content

In order to provide the computer scientist with the best content for the successful development of mobile applications, TECH has a team of professionals in programming, application development and new technologies who have selected exclusive content and who will accompany the computer scientist in the study process during the modules developed over 24 months. From an online platform, the graduate will be able to access interactive content, distributed in videos, graphics, examples based on problems, audiovisual material and a large community to discuss the issues raised.

You will have the best audiovisual content with practical exercises and interactive formats to make your learning process agile and efficient"

Module 1. Programming Methodologies in Mobile Application Development

1.1. Software Development Processes
- 1.1.1. *Waterfall*
- 1.1.2. *Spiral*
- 1.1.3. RUP
- 1.1.4. V-Model

1.2. Agile Software Development Processes
- 1.2.1. *Scrum*
- 1.2.2. XP
- 1.2.3. Kanban

1.3. Unified Modeling Language (UML)
- 1.3.1. UML
- 1.3.2. Types of Modeling
- 1.3.3. Basic Blocks of UML

1.4. UML Behavioral Diagrams
- 1.4.1. *Activity Diagram*
- 1.4.2. *Use Case Diagram*
- 1.4.3. *Interaction Overview Diagram*
- 1.4.4. *Timing Diagram*
- 1.4.5. *State Machine Diagram*
- 1.4.6. *Communication Diagram*
- 1.4.7. *Sequence Diagram*

1.5. UML Structural Diagrams
- 1.5.1. *Class Diagram*
- 1.5.2. *Object Diagram*
- 1.5.3. *Component Diagram*
- 1.5.4. *Composite Structure Diagram*
- 1.5.5. *Deployment Diagram*

1.6. Creative Design Patterns
- 1.6.1. *Singleton*
- 1.6.2. *Prototype*
- 1.6.3. *Builder*
- 1.6.4. *Factory*
- 1.6.5. *Abstract Factory*

1.7. Structural Design Patterns
- 1.7.1. *Decorator*
- 1.7.2. *Facade*
- 1.7.3. *Adapter*
- 1.7.4. *Bridge*
- 1.7.5. *Composite*
- 1.7.6. *Flyweight*
- 1.7.7. *Proxy*

1.8. Behavioral Patterns
- 1.8.1. *Chain of Responsability*
- 1.8.2. *Command*
- 1.8.3. *Iterator*
- 1.8.4. *Mediator*
- 1.8.5. *Memento*
- 1.8.6. *Observer*
- 1.8.7. *State*
- 1.8.8. *Strategy*
- 1.8.9. *Template Method*
- 1.8.10. *Visitor*

1.9. *Testing*
- 1.9.1. Unit Tests
- 1.9.2. Integration Tests
- 1.9.3. White Box Techniques
- 1.9.4. Black Box Techniques

1.10. Quality
- 1.10.1. ISO
- 1.10.2. ITIL
- 1.10.3. COBIT
- 1.10.4. PMP

Module 2. Technologies in Mobile Application Development

2.1. Mobile Devices
- 2.1.1. Mobile Devices
- 2.1.2. Infrastructure of a Mobile Device
- 2.1.3. Hardware Manufacturers
- 2.1.4. Software Developers
- 2.1.5. Service Providers
- 2.1.6. Platform Providers
- 2.1.7. Main Platforms

2.2. Physical Components of Mobile Devices
- 2.2.1. Storage
  - 2.2.1.1. Immutable
  - 2.2.1.2. Mutable
  - 2.2.1.3. Temporal
  - 2.2.1.4. External
- 2.2.2. Presenters
  - 2.2.2.1. Displays, Loudspeakers, Haptic Responses
- 2.2.3. Input Methods
  - 2.2.3.1. Buttons/Keypads
  - 2.2.3.2. Screens
  - 2.2.3.3. Microphones
  - 2.2.3.4. Movement Sensors

2.2.4. Energy sources
2.2.4.1. Sources of Energy
2.2.4.2. Adaptive Use of Resources
2.2.4.3. Efficient Programming
2.2.4.4. Sustainable Development
2.3. Processors
2.3.1. Central Processor
2.3.2. Other Abstracted Processors
2.3.3. Artificial Intelligence Processors
2.4. Information Transmitters
2.4.1. Long Range
2.4.2. Mid-Range
2.4.3. Short Range
2.4.4. Ultra Short Range
2.5. Sensors
2.5.1. Internal to the Device
2.5.2. Environmental.
2.5.3. Medical
2.6. Logic Components
2.6.1. Immutable
2.6.2. Manufacturer Mutable
2.6.3. Available to the User
2.7. Categorization
2.7.1. Laptops
2.7.2. Smartphones
2.7.2.1. Tablets
2.7.2.2. Multimedia Devices
2.7.2.3. Intelligent Complements
2.7.3. Robotic Assistants
2.8. Modes of Operation
2.8.1. Disconnected
2.8.2. Connected
2.8.3. Always Available
2.8.4. Point to Point
2.9. Interactions
2.9.1. User-Mediated Interactions
2.9.2. Supplier-Mediated Interactions
2.9.3. Devices-Mediated Interactions
2.9.4. Environmentally-Mediated Interactions
2.10. Security/Safety
2.10.1. Measures Implemented by the Manufacturer
2.10.2. Measures Implemented by Suppliers
2.10.3. User-Applied Security
2.10.4. Privacy

Module 3. Work Tools for Mobile Application Development

3.1. Environment and Tools for the Development of Applications for Mobile Devices
3.1.1. Mac OS Environment Preparation
3.1.2. Linux Environment Preparation
3.1.3. Windows Environment Preparation
3.2. Command Line
3.2.1. Command Line
3.2.2. Emulators
3.2.3. Command Interpreter
3.2.4. Folder Creation
3.2.5. File Creation
3.2.6. Navigation

3.2.7. Managing Files and Folders Using the Command Line Interface
3.2.8. Licences
3.2.9. SSH
3.2.10. Command List
3.3. Software Repository Git
3.3.1. Version Control System
3.3.2. Git
3.3.3. Settings
3.3.4. Repository
3.3.5. Branches
3.3.6. Branch Management
3.3.7. Work Flows
3.3.8. Merge
3.3.9. Commands
3.4. Web Service Version Control
3.4.1. Remote Repositories
3.4.2. Settings
3.4.3. Authentication
3.4.4. Branching of Software Fork
3.4.5. Git Clone Command
3.4.6. Repositories
3.4.7. *Github Pages*
3.5. Advanced Development Tools for Applications on Mobile Devices
3.5.1. *Postman*
3.5.2. *Visual Studio Code*
3.5.3. GUI for Databases
3.5.4. *Hosting*
3.5.5. Complementary Development Tools
3.6. Web from the Prism of Application Development for Mobile Devices
3.6.2. Protocols
3.6.3. Internet Service Provider
3.6.4. IP Addresses
3.6.5. DNS Name Services
3.7. Programming in the Development of Applications for Mobile devices
3.7.1. Programming in the Development of Applications for Mobile devices
3.7.2. Programming Paradigms
3.7.3. Programming Languages
3.8. Application Development Components for Mobile Devices
3.8.1. Variables and Constants
3.8.2. Types
3.8.3. Operators
3.8.4. Declarations
3.8.5. Loops
3.8.6. Functions and Objects
3.9. Data Structure
3.9.1. Data Structure
3.9.2. Linear Structure Types
3.9.3. Functional Structure Types
3.9.4. Tree Structure Types
3.10. Algorithms
3.10.1. Algorithms in Programming Divide and Conquer
3.10.2. Voracious Algorithms
3.10.3. Dynamic Programming

Module 4. Multi-Platform Web Development for Mobiles

4.1. Multi-Platform Web Development
- 4.1.1. Multi-Platform Web Development
- 4.1.2. Hybrid Apps vs. Native Apps
- 4.1.3. Technologies to Create Hybrid Apps

4.2. *Progressive Web Apps* (PWA)
- 4.2.1. *Progressive Web Apps* (PWA)
- 4.2.2. *Progressive Web Apps* (PWA). Features
- 4.2.3. *Progressive Web Apps* (PWA). Construction
- 4.2.4. *Progressive Web Apps* (PWA). Limitations

4.3. *Framework Ionic*
- 4.3.1. *Framework Ionic* Analysis
- 4.3.2. *Framework Ionic* Features
- 4.3.3. Building an App with *Ionic*

4.4. Web Development *Frameworks*
- 4.4.1. *Framework* Analysis in Web Development
- 4.4.2. Web Development *Frameworks*
- 4.4.3. Web *Frameworks* Comparison

4.5. Angular *Framework*
- 4.5.1. Angular *Framework*
- 4.5.2. Using Angular in multi-platform Application Development
- 4.5.3. Angular + *Ionic*
- 4.5.4. Building Apps in Angular

4.6. *React* Development Library
- 4.6.1. *JavaScript React*Library
- 4.6.2. JavaScript React Library Use
- 4.6.3. *React Native*
- 4.6.4. *React* + *Ionic*
- 4.6.5. Building Apps in *React*

4.7. Vue Development *Framework*
- 4.7.1. Vue Development *Framework*
- 4.7.2. Vue Development *Framework* Use
- 4.7.3. Vue + *Ionic*
- 4.7.4. Building Apps in Vue

4.8. Electron Development *Frameworks*
- 4.8.1. Electron Development *Frameworks*
- 4.8.2. Electronic Development *Frameworks*. Use
- 4.8.3. Deploying Our Apps Also on Desktop

4.9. *Flutter* Mobile Device Development Tool
- 4.9.1. *Flutter* Mobile Device Development Tool
- 4.9.2. Use of *Flutter* SDK
- 4.9.3. Building Apps in *Flutter*

4.10. Development Tools for Mobile Devices Comparison
- 4.10.1. Tools for Application Development on Mobile Devices
- 4.10.2. *Flutter* vs. Ionic
- 4.10.3. Selection of the Most Suitable *Stack* for Creating an App

Module 5. Database for Mobile Application Development

5.1. Databases in Mobile Devices
- 5.1.1. Data Persistence in Mobile Application Development
- 5.1.2. Database Capabilities for Mobile Applications
- 5.1.3. SQL *Structured Query Language*

5.2. Choice of Database for Mobile Applications
- 5.2.1. Database-Driven Analysis of Applications in Mobile Devices
- 5.2.2. Database Categories
- 5.2.3. Database Overview

5.3. Development with SQLite
- 5.3.1. SQLite Database
- 5.3.2. Deployment of the Model
- 5.3.3. Connection to SQLite

5.4. Development with Oracle Berkeley DB
- 5.4.1. Berkeley DB Database
- 5.4.2. Model Deployment
- 5.4.3. Connection to Berkeley DB

5.5. Development with Realm
- 5.5.1. Realm Capabilities
- 5.5.2. Database Creation in Realm
- 5.5.3. Connection to Realm

5.6. Development with CouchDB Lite
- 5.6.1. CouchDB Lite Database
- 5.6.2. Database Creation with CouchDB Lite
- 5.6.3. Connection with CouchDB Lite

5.7. Development with MySQL Centralized Database
- 5.7.1. MySQL Database
- 5.7.2. Deployment of Relational Model with MySQL
- 5.7.3. Connection to MySQL

5.8. Centralized Developments Oracle, MS SQL Server, MongoDB
- 5.8.1. Development with Oracle
- 5.8.2. Development with MS SQL Server
- 5.8.3. Development with MongoDB

5.9. Graph Type Data
- 5.9.1. Graph Oriented Database
- 5.9.2. Database Creation with Neo4j
- 5.9.3. Connection to Neo4j from the Mobile App

5.10. Environments with Storage Capacities
- 5.10.1. Firebase Developments
- 5.10.2. Core Data Developments
- 5.10.3. Visual Builder Cloud Service Development

Module 6. Application Development for iOS Systems

6.1. Xcode Development Environment
- 6.1.1. Creation of a Project
- 6.1.2. Configuration of an Emulator for Compiling
- 6.1.3. Configuration of a Physical Phone for Compiling

6.2. Swift Programming Language
- 6.2.1. Swift I: Programming Language
- 6.2.2. Swift II: Functions and Loops
- 6.2.3. Swift III: Lambdas and Structs

6.3. Libraries and Cocoa Pods
- 6.3.1. Pods: Installation
- 6.3.2. Configuration of Cocoa Pods
- 6.3.3. Structure of Cocoa Pods

6.4. Libraries: Api, Database and R.swift
- 6.4.1. Alamofire
- 6.4.2. SQL Databases with GRDB
- 6.4.3. R.swift

6.5. Screen Design
- 6.5.1. Design with Storyboard
- 6.5.2. Responsive Design
- 6.5.3. View Design by Code and SwiftUI

6.6. View Setup
- 6.6.1. UIViewController and Its Lifecycle
- 6.6.2. Interaction between Different Screens
- 6.6.3. Types of Transitions and Modes

6.7. Sensors and Localization
- 6.7.1. Access to Sensors
- 6.7.2. Access to Foreground Localization
- 6.7.2. Background Location Access

6.8. Architecture
- 6.8.1. MVP
- 6.8.2. VIPER
- 6.8.3. IOS Development Architecture

6.9. Monetization and *Analytics*
- 6.9.1. *Firebase Analytics*
- 6.9.2. *Firebase Crashlytics*
- 6.9.3. Monetization and Ads with Google ADMob

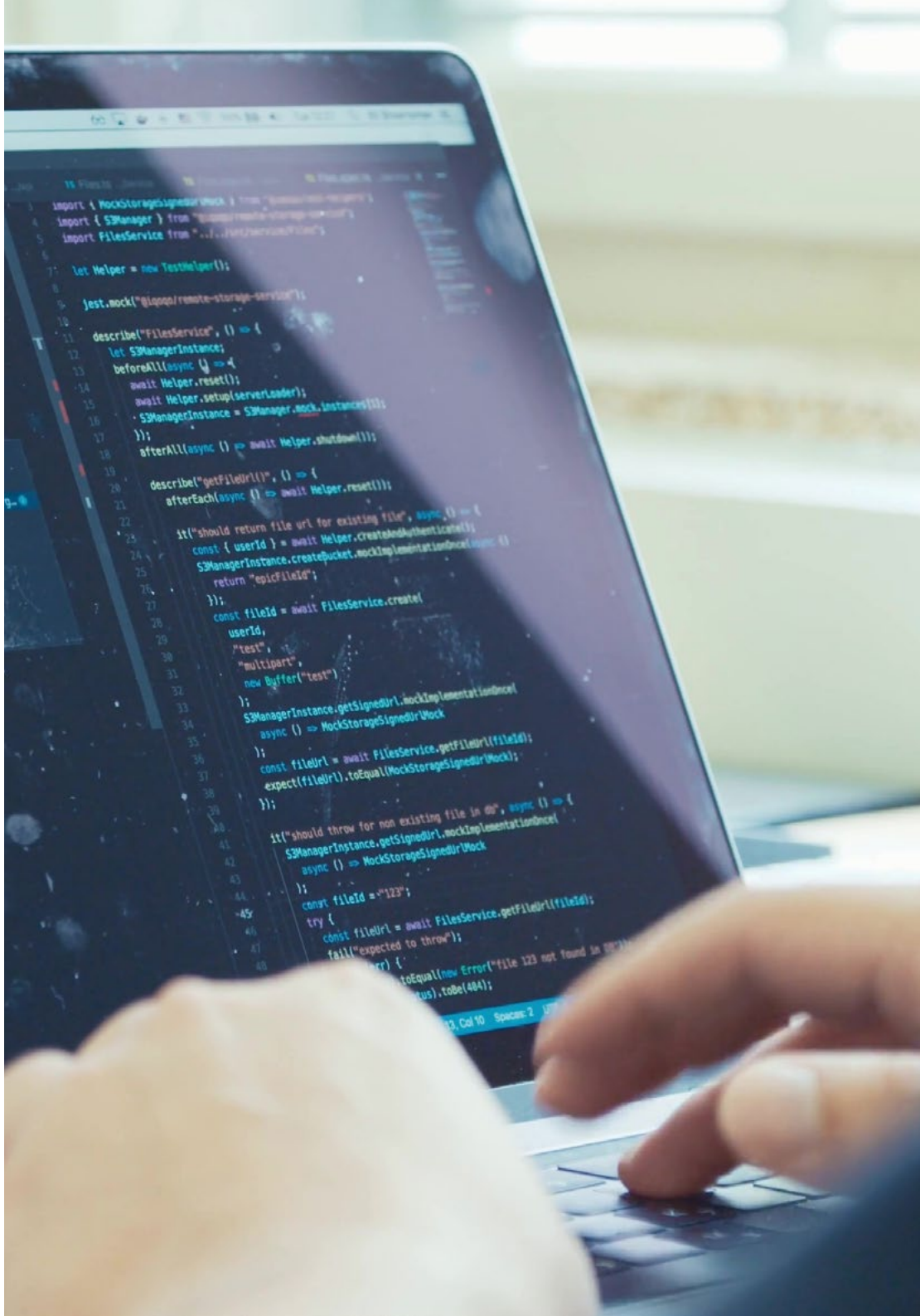
6.10. App Store and Versioning
- 6.10.1. Configuration of an App Store Account
- 6.10.2. *Test Flight* Versions
- 6.10.3. Launch into Production

Module 7. Continuous Integration Deployments for Mobiles

7.1. DevSecOps
- 7.1.1. DevSecOps Use
- 7.1.2. Static Analyzers
- 7.1.3. Dynamic Analysis Safety Tests

7.2. Continuous Monitoring
- 7.2.1. Continuous Monitoring
- 7.2.2. Continuous Monitoring Analysis and Advantages
- 7.2.3. Continuous Monitoring Platforms

7.3. Implementation
- 7.3.1. Local Machine Implementation
- 7.3.2. Shared Machine Implementation
- 7.3.3. Cloud-Based Implementation
- 7.3.4. Configuration Management



Module 8. Mobile *User Experience*

8.1. *User Experience*
- 8.1.1. *Client Experience*
- 8.1.2. *Client Experience*. Requirements
- 8.1.3. Bidirectionality with the Client

8.2. *Client Experience* Objectives and Equipment
- 8.2.1. *Client Experience*. Objectives and Equipment
- 8.2.2. Iterative Processes
- 8.2.3. Information Required

8.3. Micro-Interactions
- 8.3.1. *End-to-End* Relationship
- 8.3.2. Interactions
- 8.3.3. Omnichannel

8.4. User Behavior
- 8.4.1. Foundation Design
- 8.4.2. Web and Session Analytics
- 8.4.3. Analytics Experts

8.5. State of the Art Technology
- 8.5.1. *Machine Learning*
- 8.5.2. Blockchain
- 8.5.3. Internet of Things

8.6. Technical Components
- 8.6.1. Technical Components
- 8.6.2. Advanced Components: Devices
- 8.6.3. Advanced Components: Different Profiles

8.7. Usability
- 8.7.1. Nielsen Heuristics
- 8.7.2. User Tests
- 8.7.3. Usabilidad. Errors

8.8. UX Techniques *User Experience*
- 8.8.1. Rules
- 8.8.2. *Prototyping*
- 8.8.3. *Low-Code* Tools

8.9. Visual Strategy
- 8.9.1. *User Interface* Designer
- 8.9.2. *User Interface* Work on the Web
- 8.9.3. *User Interface* Work in Applications

8.10. *Developer Frameworks*
- 8.10.1. CX *Frameworks*
- 8.10.2. UX *Frameworks*
- 8.10.3. UI *Frameworks*

Module 9. Security on Mobile Devices

9.1. Mobile Device Security Architecture
- 9.1.1. Physical Security of Devices
- 9.1.2. Operating System Security
- 9.1.3. Application Security
- 9.1.4. Data Security
- 9.1.5. Communications Security
- 9.1.6. Security of Enterprise Devices

9.2. Securing Mobile Hardware
- 9.2.1. Mobile Devices
- 9.2.2. Wearable Devices
- 9.2.3. Automotive
- 9.2.4. IOT Devices
- 9.2.5. TV Devices

9.3. Operating System Security
- 9.3.1. Android Mobile Devices
- 9.3.2. Apple IOS Mobile Devices
- 9.3.3. Other Existing Mobile Devices: Blackberry, etc
- 9.3.4. Wearable Devices
- 9.3.5. Automotive Operating Systems
- 9.3.6. Mobile Devices in the *Internet of Things* (IoT)
- 9.3.7. *SmartTV* Devices

9.4. Securing Mobile Applications
- 9.4.1. Android Mobile Devices
- 9.4.2. Apple IOS Mobile Devices
- 9.4.3. Other Mobile Devices Blackberry
- 9.4.4. *Wearables* Devices
- 9.4.5. Automotive Operating Systems
- 9.4.6. Mobile Devices in the *Internet of Things* (IoT)
- 9.4.7. *SmartTV* Devices

9.5. Securing Data in Mobile Applications
- 9.5.1. Android Mobile Devices
- 9.5.2. Apple IOS Mobile Devices
- 9.5.3. Other Mobile Devices Blackberry
- 9.5.4. *Wearables* Devices
- 9.5.5. Automotive Operating Systems
- 9.5.6. Mobile Devices in the *Internet of Things* (IoT)
- 9.5.7. *SmartTV* Devices

9.6. Mobile *Market Places* Security
- 9.6.1. *Google Play* by Google
- 9.6.2. *Play Store* by Apple
- 9.6.3. Other *Market Places*
- 9.6.4. *Rooting* of Mobile Devices

9.7. Multi-Platform Security Solutions
- 9.7.1. *Mobile Device Management* (MDM)
- 9.7.2. Types of Solutions on the Market
- 9.7.3. Securing Devices Using MDM (*Master Data Management*)

9.8. Secure Mobile Application Development
- 9.8.1. Use of Patterns for Safe Development
- 9.8.2. Integrated Security Test Management
- 9.8.3. Secure Application Deployment

9.9. Permission Management in Mobile Devices
- 9.9.1. Permission Systems
- 9.9.2. Execution of Processes in the Core
- 9.9.3. Execution Threads and Events

9.10. Security Recommendations for Mobile Devices
- 9.10.1. NSA Recommendations on Mobile Devices
- 9.10.2. INCIBE Recommendations on Mobile Devices
- 9.10.3. ISO 27001:2013 Annex

Module 10. Android Programming Language

10.1. Android Platform
- 10.1.1. Android Platform
- 10.1.2. Android Operating System
- 10.1.3. *Open Handset Alliance* in Android development

10.2. Android Architecture
- 10.2.1. Architectural Elements of an Android System
- 10.2.2. Communication between Elements
- 10.2.3. Extensibility of the Android Architecture
- 10.2.4. Machine Resource Management: Battery and Memory
- 10.2.5. Android Emulators

10.3. Android Linux Core
- 10.3.1. Composition of the Core
- 10.3.2. Structural Elements of the Core
- 10.3.3. Dalvik Virtual Machine
- 10.3.4. The Android RunTime Virtual Machine (ART)

10.4. Native Android Libraries
- 10.4.1. Native Android Libraries
- 10.4.2. *Support Library*
- 10.4.3. Native Libraries and Extensibility

10.5. The Android File and Data System
- 10.5.1. Structure of a Typical Android Application
- 10.5.2. YAFFS2 and ext4 File System
- 10.5.3. Use of SQLite and Room for Data Management

10.6. Android Security
- 10.6.1. Permission System
- 10.6.2. Digital Signatures in the *Android Application Package* (apk)
- 10.6.3. Execution of Processes in the Core
- 10.6.4. Execution Threads and Events

10.7. Structural Components of a Standard Application
- 10.7.1. *View*
- 10.7.2. *Activity*
- 10.7.3. *Fragment*
- 10.7.4. *Service*
- 10.7.5. *Intent*
- 10.7.6. *Broadcasts Receiver* and *Content Provider*
- 10.7.7. Data Management and User Preferences

10.8. Android Versions
- 10.8.1. Android Versions
- 10.8.2. Deployment of Android Versions
- 10.8.3. Dispersion of Android Distributions
- 10.8.4. Android vs. Apple IOS and Other Mobile Systems

10.9. Android for Vehicles
- 10.9.1. Android and the Automotive World
- 10.9.2. Structural Elements in an Automotive Android System
- 10.9.3. Communication between Devices

10.10. Android in Home Automation, *Wearables* and *Internet of Things* (IoT)
- 10.10.1. The Connected World
- 10.10.2. Structural Elements in an Android Home Automation System
- 10.10.3. Elements of Android *Wearable*
- 10.10.4. Android in the *Internet of Things* (IoT)

Module 11. *Frameworks* in Android Application Development

11.1. *Frameworks* in Android Application Development
- 11.1.1. *Frameworks* in Android Application Development
- 11.1.2. *Frameworks* Typology
- 11.1.3. Choice of the *Framework* for the project

11.2. Android *Framework* Implementation
- 11.2.1. Android Frameworks *Core* for Java/Kotlin
- 11.2.2. *Jetpack Compose*
- 11.2.3. *Frameworks* in Other Languages

11.3. Library Management Systems in Development
- 11.3.1. *Gradle*
- 11.3.2. Automation with *Gradle*
- 11.3.3. Maven Development Tool

11.4. Clean Code
- 11.4.1. Ordered Code
- 11.4.2. Code Preparation in Android Applications
- 11.4.3. *Bikeshedding* and Prioritization

11.5. Android Development Patterns
- 11.5.1. Pattern Categories
- 11.5.2. Differences between Patterns
- 11.5.3. *Factory, Observer* and *Singleton*

11.6. MVP Model, Viewer and Presenter
- 11.6.1. MVC: Model, View and Controller
- 11.6.2. Model, Viewer and Presenter
- 11.6.3. Practical Example: Pokemon Battle

11.7. MVVM: Model, View and *View Model*
- 11.7.1. MVC vs. MVVM
- 11.7.2. Model, View and *View Model*
- 11.7.3. Practical Example: Pokemon Battle II

11.8. *Frameworks* and Libraries Most Used in Android
- 11.8.1. API Interaction Libraries
- 11.8.2. Data Conversion Libraries
- 11.8.3. *Firebase* and *Firebase Analytics*

11.9. Android's Visual *Framework*
- 11.9.1. Life Cycle of an Android Application
- 11.9.2. XML View Design
- 11.9.3. Design of Elements and Animations in XML

11.10. Android *Frameworks* in Other Languages
- 11.10.1. *React Native*
- 11.10.2. *Flutter*
- 11.10.3. *Ionic*

Module 12. Interfaces and *Layouts* in Android Application Development

12.1. Android Interface Lifecycle
- 12.1.1. Android Lifecycle
- 12.1.2. Process-Activity Relationship
- 12.1.3. Application State Persistence
- 12.1.4. *Clean* Architecture Applied to Android

12.2. Views in Android Application Development (*Views*)
- 12.2.1. *Clean* Architecture Presentation Layer
- 12.2.2. *Recycler View*
- 12.2.3. *Adapter View*

12.3. Designs in Android Application Development (*Layouts*)
- 12.3.1. *Layouts* in Android
- 12.3.2. *Constraint Layout*
- 12.3.3. Creating *Layouts* Using *Android Studio Layout Editor*

12.4. Animations in Android Application Development (*Animations*)
- 12.4.1. Icons and Images
- 12.4.2. Transitions
- 12.4.3. Difference between Property Animation and View Animation

12.5. Activities and Intentions in Android Application Development (*Activity* and *Intentions*)
- 12.5.1. Explicit and Implicit Intentions
- 12.5.2. Action Bar
- 12.5.3. Communication between Activities

12.6. Alternative and System Resources (*Material Design, Cardboard,* etc.)
- 12.6.1. Material *Design* for Android
- 12.6.2. Multimedia in Android
- 12.6.3. Virtual Reality with *Google Cardboard for Android* NDK

12.7. Styles and Themes in Android Application Development
- 12.7.1. Styles in an Android Project
- 12.7.2. Themes for the Android Project
- 12.7.3. Reuse of Styles and Themes

12.8. Graphics, Touch Screen and Sensors
- 12.8.1. Working with Advanced Graphics
- 12.8.2. Management of Touch Screen and Keypad Devices
- 12.8.3. Use of Android Device Sensors

12.9. Augmented Reality Designs
- 12.9.1. Complex Interfaces Using the Camera
- 12.9.2. Position Sensors and GPS in Augmented Reality
- 12.9.3. Presentation on Non-Standard Screens
- 12.9.4. Common Errors and Problems

12.10. Advanced Interface Configuration with *AndroidManifest.xml*
- 12.10.1. The Power of the Android Manifest File
- 12.10.2. Programmatic vs. Declarative Design
- 12.10.3. Key Components of the File

Module 13. Programming Language in Android Applications Kotlin

13.1. Kotlin Programming Language
- 13.1.1. Kotlin Programming Language
- 13.1.2. Kotlin Programming Language. Features
- 13.1.3. Running a Program in Kotlin

13.2. Programming in Kotlin
- 13.2.1. Structure of a Program in Kotlin
- 13.2.2. Reserved Words and Syntax
- 13.2.3. Write through Console and Read User *Inputs*-Hello World

13.3. Types and Variables in Kotlin
- 13.3.1. Types and Variables in Kotlin
- 13.3.2. Variable Declaration: Var vs. Val
- 13.3.3. Operators
- 13.3.4. Type Conversion
- 13.3.5. *Arrays*

13.4. Flow Control in Kotlin
- 13.4.1. Flow Control
- 13.4.2. Conditional Expressions
- 13.4.3. Loops

13.5. Functions in Kotlin
- 13.5.1. Functions in Kotlin
- 13.5.2. Structure of a Function
- 13.5.3. *Scope Functions*

13.6. Types and Objects in Kotlin
- 13.6.1. Types and Objects in Kotlin
- 13.6.2. Classes
- 13.6.3. Objects
- 13.6.4. Constructors and Property Initialization
- 13.6.5. Nested Classes and Inner Classes
- 13.6.6. Data Classes

13.7. Kotlin Inheritance
- 13.7.1. Heritage
- 13.7.2. Superclasses and Subclasses
- 13.7.3. Overwriting Properties and Functions
- 13.7.4. Inheritance vs. Other Types of Relationship between Classes
- 13.7.5. Sealed Classes
- 13.7.6. Listed

13.8. Abstract Classes and Interfaces in Kotlin
- 13.8.1. Abstract Classes and Interfaces
- 13.8.2. Abstract Classes
- 13.8.3. Interfaces
- 13.8.4. Validation and Conversion of Operator-Types *Is, When, As*

13.9. Kotlin Collections
- 13.9.1. Kotlin Collections
- 13.9.2. List
- 13.9.3. Set
- 13.9.4. Map

13.10. Exception and Null Value Handling in Kotlin
- 13.10.1. Exception and Null Value Handling
- 13.10.2. Null Value, *Nullable* and *Non-Nullable* Types
- 13.10.3. Exceptions

Module 14. Programming Language in Android Applications Advanced Kotlin Genericity, Functional Programming and Parallelism

14.1. Genericity in Kotlin
- 14.1.1. Genericity in Kotlin
- 14.1.2. Genericity in Collections, Functions, Classes and Interfaces
- 14.1.3. Covariance and Contravariance: *Out* or *In*

14.2. Lambda Functions in Kotlin
- 14.2.1. Lambda Functions
- 14.2.2. Structure of a Lambda Function
- 14.2.3. Use of Lambda Functions

14.3. Higher Order Functions in Kotlin
- 14.3.1. Higher Order Functions
- 14.3.2. Standard Kotlin Higher-Order Functions
- 14.3.3. Linking Function Calls

14.4. Kotlin Extensions
- 14.4.1. Kotlin Extensions
- 14.4.2. Extension Functions
- 14.4.3. Extension Properties
- 14.4.4. Accompanying Objects

14.5. *Delegation* Pattern in Kotlin
- 14.5.1. *Delegation* Pattern
- 14.5.2. *Delegation* in Kotlin
- 14.5.3. Delegated Properties

14.6. Annotations and Reflection in Kotlin
- 14.6.1. Annotations and Reflection
- 14.6.2. Annotations in Kotlin
- 14.6.3. Reflection in Kotlin

14.7. *Testing* in Kotlin
- 14.7.1. *Testing* in Kotlin
- 14.7.2. Kotlin *Testing Frameworks* and Libraries
- 14.7.3. Kotest

14.8. Asynchronous Programming in Kotlin
- 14.8.1. Asynchronous Programming
- 14.8.2. Asynchronous Programming Techniques in Kotlin
- 14.8.3. Comparative Programming Techniques

14.9. Coroutine in Kotlin
- 14.9.1. Coroutines
- 14.9.2. Channels
- 14.9.3. Context and *Dispatchers*
- 14.9.4. Shared State and Concurrency
- 14.9.5. Exception Handling in Coroutines

14.10. Kotlin Ecosystem
- 14.10.1. Kotlin Ecosystem
- 14.10.2. Libraries for Kotlin
- 14.10.3. Tools for Kotlin

Module 15. Data Management on Android Devices

15.1. Data Management Typology
- 15.1.1. Data Management in Mobile Devices
- 15.1.2. Alternatives for Data Management in Android devices
- 15.1.3. Data Generation for Work with Artificial Intelligence and Usage Analytics
- 15.1.4. Performance Measurement Tools for Optimal Data Management

15.2. User Preferences Management
- 15.2.1. Types of Data Involved in Preference Files
- 15.2.2. User Preferences Management
- 15.2.3. Exporting Preferences: Permissions Management

15.3. File Storage System
- 15.3.1. File System Classification on Mobile Devices
- 15.3.2. Internal File System
- 15.3.3. External File System

15.4. JSON Files as Storage in Android
- 15.4.1. Unstructured Information in JSON Files
- 15.4.2. JSON Data Management Libraries
- 15.4.3. Use of JSON in Android: Recommendations and Optimizations

15.5. XML Files as Android Storage
- 15.5.1. XML Format in Android
- 15.5.2. XML through SAX Libraries
- 15.5.3. XML through DOM Libraries

15.6. SQLite Database
- 15.6.1. Relational Database for Data Management
- 15.6.2. Database Use
- 15.6.3. SQLite Methods for Data Management

15.7. Advanced Use of SQLite Databases
- 15.7.1. Failure Recovery Using SQLite Transactions
- 15.7.2. Use of Caching to Accelerate Data Access
- 15.7.3. Mobile Databases

15.8. Room Library
- 15.8.1. Architecture of the Room Library
- 15.8.2. Room Library: Functionality
- 15.8.3. Room Library: Advantages and Disadvantages.

15.9. *Content Provider* to Share Information
- 15.9.1. *Content Provider* to Share Information
- 15.9.2. Content Provider in Android Technician Use
- 15.9.3. *Content Provider* Security

15.10. Internet Cloud Data Collection
- 15.10.1. Android and *Cloud* Storage Systems
- 15.10.2. SOAP and REST Services for Android
- 15.10.3. Problems of Distributed Systems
- 15.10.4. Internet as a Backup of Application Data

Module 16. Android Device Tools

16.1. Management: "TO DO" Tools
- 16.1.1. Market Tools
- 16.1.2. Market Tools: Comparison of Functionalities
- 16.1.3. Management Tools: Differences

16.2. MDM: Enterprise Mobile Device Management
- 16.2.1. Control of Enterprise Devices
- 16.2.2. Analysis of the Main Market Tools
- 16.2.3. Choosing the Tool

16.3. CRM: Market Tools
- 16.3.1. Analysis of Market Tools with Android Application
- 16.3.2. Market Tools Efficiency
- 16.3.3. Market Tools Uses

16.4. Android Drones
- 16.4.1. Apps for Android Devices for Drone Control
- 16.4.2. Autonomous Controls
- 16.4.3. Drone Uses in Android

16.5. Android, Added Value in Banking Platforms
- 16.5.1. Android in Banking Platforms
- 16.5.2. Risks and Fraud of Cybercriminals
- 16.5.3. Uses of Mobile Devices

16.6. *Brokering* in Mobile Devices
- 16.6.1. Market Tools and Their Use
- 16.6.2. Comparison of Tools
- 16.6.3. Choice of Tool for Each Use

16.7. Entertainment and Training Tools
- 16.7.1. Uses
- 16.7.2. Market Tools
- 16.7.3. Feature Comparisons between Android Development Tools

16.8. IoT Android
- 16.8.1. *Framework* and Market Platforms
- 16.8.2. Android IoT Risks and Considerations
- 16.8.3. IoT Uses in Android

16.9. Process Efficiency
- 16.9.1. Analysis of Market Tools for Creating Apps
- 16.9.2. Comparison of Android App Creation Tools
- 16.9.3. *Use Case*

16.10. Most Downloaded Applications at Present
- 16.10.1. Most Downloaded Tools at Present
- 16.10.2. Grouping by Families
- 16.10.3. Primary, Secondary and Comparative Uses with IOS

Module 17. *Responsive*Design in Android

17.1. *Responsive Design*
- 17.1.1. *Responsive* Design
- 17.1.2. Usability, Accessibility and UX
- 17.1.3. *Responsive* Design Advantages and Disadvantages.

17.2. *Mobile vs Tablet vs Web vs Smartwatches*
- 17.2.1. Different Formats, Different Sizes, Different Needs
- 17.2.2. Design Problems
- 17.2.3. Adaptive vs. Responsive

17.3. Style Guide
- 17.3.1. Style Guide Uses
- 17.3.2. *Design* Meterial
- 17.3.3. Own Style Guide

17.4. Flexible *Layouts*
- 17.4.1. Flexible Layouts
- 17.4.2. Basic Layouts
- 17.4.3. Layouts in *Grid*
- 17.4.4. *Layouts* with *Relative Layout*
- 17.4.5. *Layouts* with *Constraint Layout*

17.5. Flexible Resources
- 17.5.1. Flexible Resources
- 17.5.2. Images
- 17.5.3. 9Patch
- 17.5.4. Global Resources

17.6. Flexible Navigation
- 17.6.1. Flexible Navigation
- 17.6.2. Navigation with *Activities*
- 17.6.3. Navigation with *Fragments*

17.7. External Tools
- 17.7.1. Automatic Generators
- 17.7.2. Prototyping Tools
- 17.7.3. Design Tools

17.8. *Debug* and *Tests*
- 17.8.1. *Debug Layouts*
- 17.8.2. Automatic *Tests*
- 17.8.3. Component-Based Development
- 17.8.4. *Testing* and Trials Best Practices

17.9. Alternatives to Native Android I. *Web Pages*
- 17.9.1. Design in a *WebView*
- 17.9.2. *ChromeCustomTabs*
- 17.9.3. *Debug* and *Tests* in *Web Pages*

17.10. Alternatives to Native Android II. Hybrid Applications
- 17.10.1. *React/React Native*
- 17.10.2. *Flutter*
- 17.10.3. *Ionic*
- 17.10.4. Apache Cordova

Module 18. Marketing in Android Applications

18.1. From *Customer Service* to *Customer Experience*
- 18.1.1. *Customer Service*. Current Customer Development
- 18.1.2. User with Access to Information: Requirements and Needs
- 18.1.3. Feedback as a Source of Knowledge

18.2. *Customer Journey*
- 18.2.1. User Pathway to Conversion
- 18.2.2. Micro-Segmentation
- 18.2.3. Cross-Channel Experience

18.3. User Experience Measurement
- 18.3.1. Web and Mobile Architecture
- 18.3.2. Session Analytics as a New Standard
- 18.3.3. State of the Art of User Experience

18.4. Android Applications Marketing
- 18.4.1. CX+AI
- 18.4.2. CX+Blockchain
- 18.4.3. CX+IoT

18.5. CX Products (Customer Experience)
- 18.5.1. Industry Standards
- 18.5.2. Telepresence
- 18.5.3. Customer Experience for all Development Agents

18.6. User-Centered Work
- 18.6.1. Equipment
- 18.6.2. Designer Thinking
- 18.6.3. Field Work

18.7. User Science
- 18.7.1. User Science Golden Rules
- 18.7.2. Iteration
- 18.7.3. Common Errors

18.8. Prototyping and *Wireframing*
- 18.8.1. Prototyping and *Wireframing*
- 18.8.2. *Hands-on*
- 18.8.3. Advanced Level

18.9. Mobiles Interfaces
- 18.9.1. Visual Design Rules
- 18.9.2. App Interface Keys
- 18.9.3. Best Practices in the Development of Mobile Interfaces

18.10. Best Practices in Users Experience. Tips for Developers
- 18.10.1. Level One Good Practices in CX
- 18.10.2. Level Two Good Practices in UX
- 18.10.3. Level Three Good Practices in UI

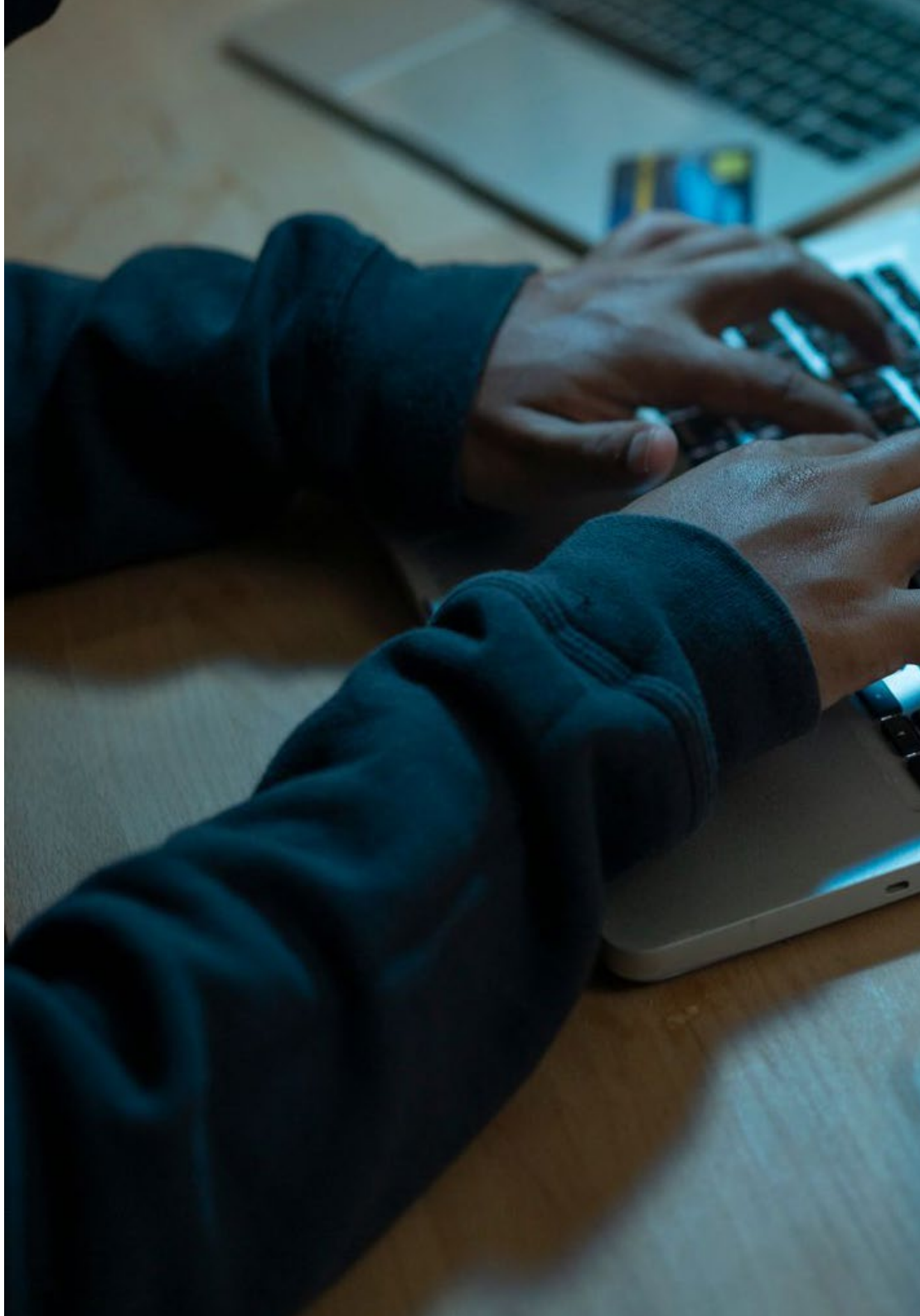
Module 19. Android Application Life Cycle: Cloud, Playstore and Versioning

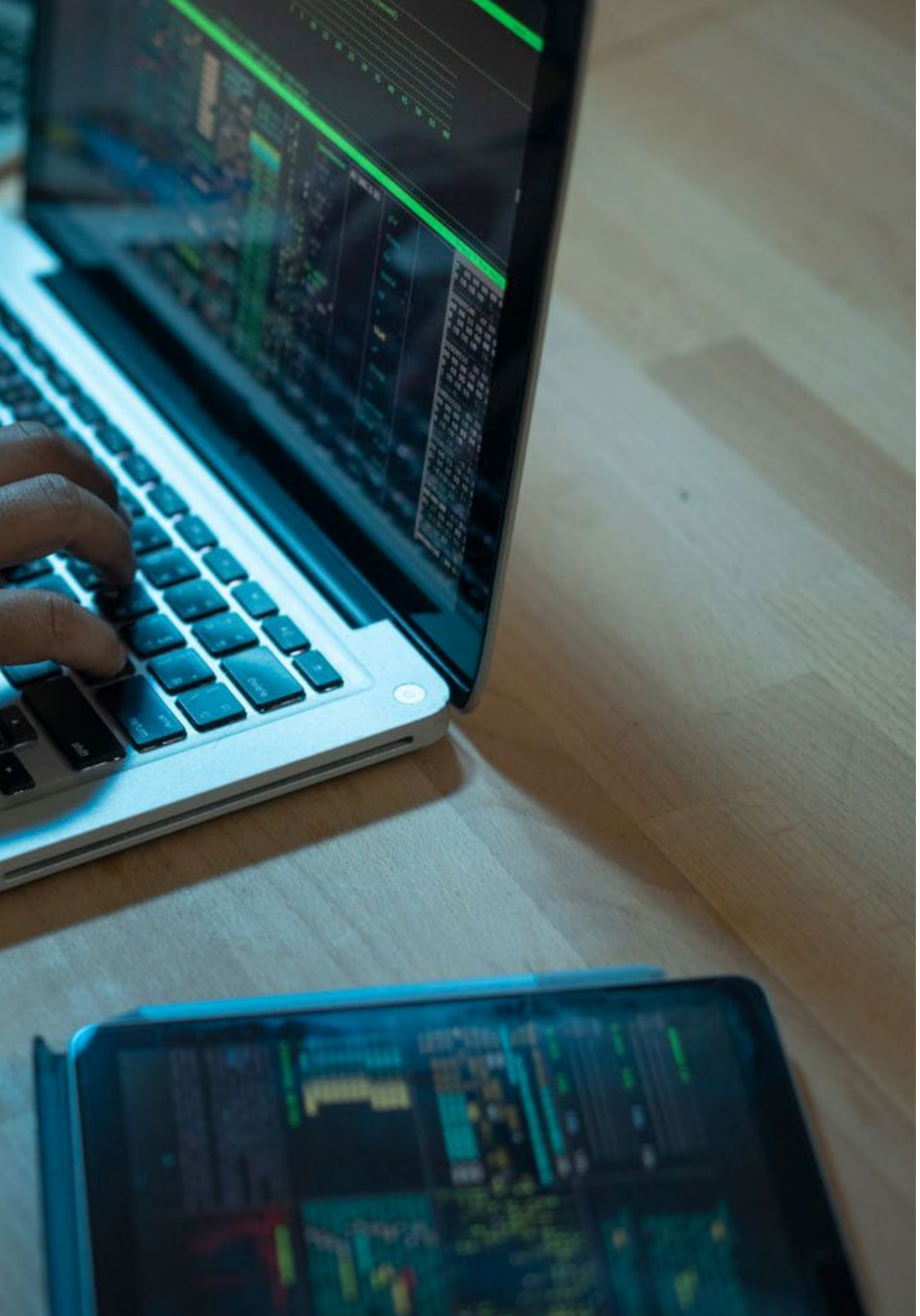
19.1. *Software* Life Cycle
- 19.1.1. *Software* Life Cycle
- 19.1.2. Agile Methodologies
- 19.1.3. The Continuous Agile *Software* Cycle

19.2. Manual Product Development
- 19.2.1. Manual Integration
- 19.2.2. Manual Delivery
- 19.2.3. Manual Deployment

19.3. Supervised Integration
- 19.3.1. Continuous Integration
- 19.3.2. Manual Revision
- 19.3.3. Static Automatic Revisions

19.4. Logical Tests
- 19.4.1. Unit Tests
- 19.4.2. Integration Tests
- 19.4.3. Behavior Tests





19.5. Continuous Integration
- 19.5.1. Continuous Integration Cycle
- 19.5.2. Dependencies between Integrations
- 19.5.3. Continuous Integration as a Repository Management Methodology

19.6. Continuous Delivery
- 19.6.1. Continuous Delivery: Types of Problems to Be Solved
- 19.6.2. Continuous Delivery: Problem Solving
- 19.6.3. Advantages of Continuous Delivery

19.7. Continuous Deployment
- 19.7.1. Continuous Deployment Types of Problems to Be Solved
- 19.7.2. Continuous Deployment Problem Solving

19.8. *Firebase Test Lab*
- 19.8.1. Configuration from *GCloud*
- 19.8.2. Jenkins Configuration
- 19.8.3. Use of Jenkins: Advantages

19.9. *Gradle* Configuration
- 19.9.1. *Gradle* Automation System
- 19.9.2. *Gradle Build Flavors* Component
- 19.9.3. *Gradle Lint* Component

19.10. Android Application Life Cycle. Example
- 19.10.1. *SemaphoreCI* and *GitHub* Configuration
- 19.10.2. Configuration of Work Blocks
- 19.10.3. Promotions and *Deployment*

Now the road to a better future begins and you are part of it. Start designing the best mobile applications with this Advanced Master's Degree"

06

Methodology

This academic program offers students a different way of learning. Our methodology uses a cyclical learning approach: **Relearning.**

This teaching system is used, for example, in the most prestigious medical schools in the world, and major publications such as the **New England Journal of Medicine** have considered it to be one of the most effective.

Discover Relearning, a system that abandons conventional linear learning, to take you through cyclical teaching systems: a way of learning that has proven to be extremely effective, especially in subjects that require memorization"

Case Study to contextualize all content

Our program offers a revolutionary approach to developing skills and knowledge. Our goal is to strengthen skills in a changing, competitive, and highly demanding environment.

At TECH, you will experience a learning methodology that is shaking the foundations of traditional universities around the world"

You will have access to a learning system based on repetition, with natural and progressive teaching throughout the entire syllabus.

The student will learn to solve complex situations in real business environments through collaborative activities and real cases.

A learning method that is different and innovative

This TECH program is an intensive educational program, created from scratch, which presents the most demanding challenges and decisions in this field, both nationally and internationally. This methodology promotes personal and professional growth, representing a significant step towards success. The case method, a technique that lays the foundation for this content, ensures that the most current economic, social and professional reality is taken into account.

> *Our program prepares you to face new challenges in uncertain environments and achieve success in your career"*

The case method has been the most widely used learning system among the world's leading Information Technology schools for as long as they have existed. The case method was developed in 1912 so that law students would not only learn the law based on theoretical content. It consisted of presenting students with real-life, complex situations for them to make informed decisions and value judgments on how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

What should a professional do in a given situation? This is the question that you are presented with in the case method, an action-oriented learning method. Throughout the course, students will be presented with multiple real cases. They will have to combine all their knowledge and research, and argue and defend their ideas and decisions.

Relearning Methodology

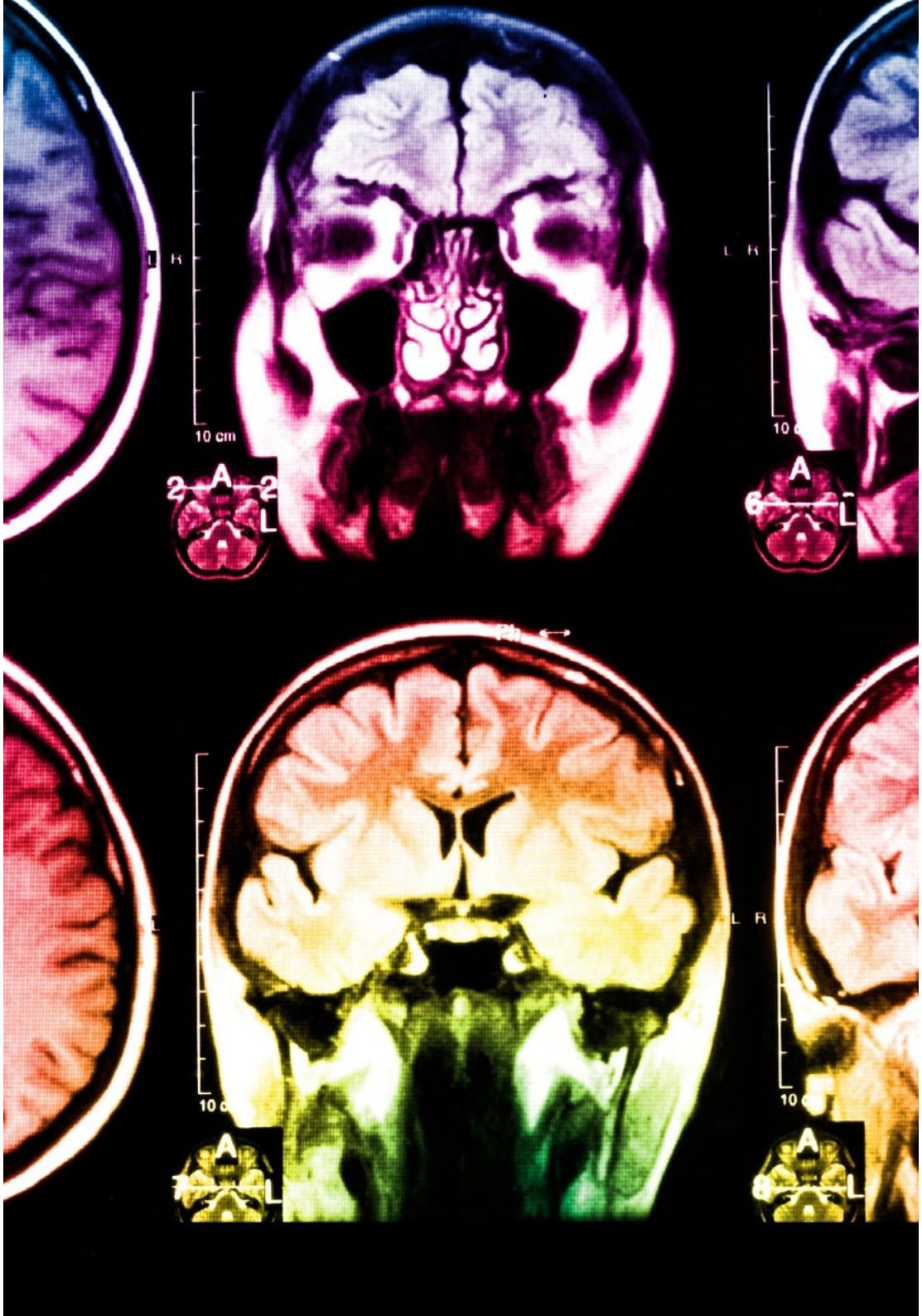
TECH effectively combines the Case Study methodology with a 100% online learning system based on repetition, which combines different teaching elements in each lesson.

We enhance the Case Study with the best 100% online teaching method: Relearning.

In 2019, we obtained the best learning results of all online universities in the world.

At TECH, you will learn using a cutting-edge methodology designed to train the executives of the future. This method, at the forefront of international teaching, is called Relearning.

Our university is the only one in the world authorized to employ this successful method. In 2019, we managed to improve our students' overall satisfaction levels (teaching quality, quality of materials, course structure, objectives...) based on the best online university indicators.



In our program, learning is not a linear process, but rather a spiral (learn, unlearn, forget, and re-learn). Therefore, we combine each of these elements concentrically. This methodology has trained more than 650,000 university graduates with unprecedented success in fields as diverse as biochemistry, genetics, surgery, international law, management skills, sports science, philosophy, law, engineering, journalism, history, and financial markets and instruments. All this in a highly demanding environment, where the students have a strong socio-economic profile and an average age of 43.5 years.

Relearning will allow you to learn with less effort and better performance, involving you more in your training, developing a critical mindset, defending arguments, and contrasting opinions: a direct equation for success.

From the latest scientific evidence in the field of neuroscience, not only do we know how to organize information, ideas, images and memories, but we know that the place and context where we have learned something is fundamental for us to be able to remember it and store it in the hippocampus, to retain it in our long-term memory.

In this way, and in what is called neurocognitive context-dependent e-learning, the different elements in our program are connected to the context where the individual carries out their professional activity.

This program offers the best educational material, prepared with professionals in mind:

Study Material

All teaching material is produced by the specialists who teach the course, specifically for the course, so that the teaching content is highly specific and precise.

These contents are then adapted in audiovisual format, to create the TECH online working method. All this, with the latest techniques that offer high-quality pieces in each and every one of the materials that are made available to the student.

Classes

There is scientific evidence suggesting that observing third-party experts can be useful.

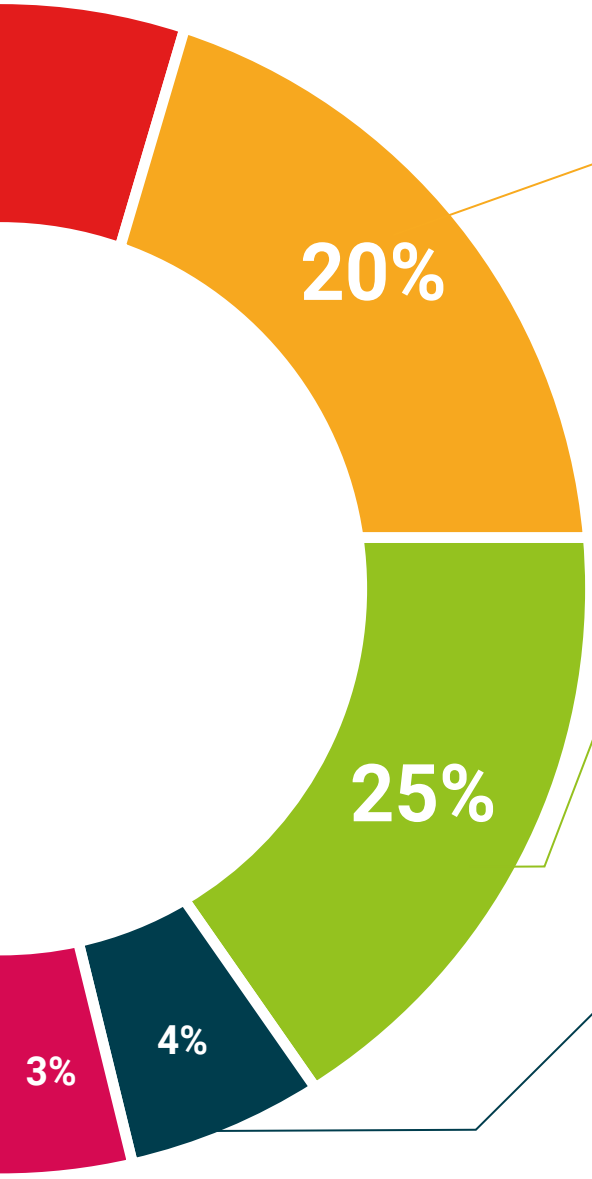
Learning from an Expert strengthens knowledge and memory, and generates confidence in future difficult decisions.

Practising Skills and Abilities

They will carry out activities to develop specific competencies and skills in each thematic area. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop in the context of the globalization that we are experiencing.

Additional Reading

Recent articles, consensus documents and international guidelines, among others. In TECH's virtual library, students will have access to everything they need to complete their course.



Case Studies

Students will complete a selection of the best case studies chosen specifically for this program. Cases that are presented, analyzed, and supervised by the best specialists in the world.

Interactive Summaries

The TECH team presents the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

This exclusive educational system for presenting multimedia content was awarded by Microsoft as a "European Success Story".

Testing & Retesting

We periodically evaluate and re-evaluate students' knowledge throughout the program, through assessment and self-assessment activities and exercises, so that they can see how they are achieving their goals.

07

Certificate

This Advanced Master's Degree in Mobile Application Development, Android Expert guarantees students, in addition to the most rigorous and up-to-date education, access to an Advanced Master's Degree issued by TECH Technological University.

Successfully complete this program and receive your university qualification without having to travel or fill out laborious paperwork"

This **Advanced Master's Degree in Mobile Application Development, Android Expert** contains the most complete and up-to-date program on the market.

After the student has passed the assessments, they will receive their corresponding **Advanced Master's Degree** diploma issued by **TECH Technological University** via tracked delivery*.

The diploma issued by **TECH Technological University** will reflect the qualification obtained in the Advanced Master's Degree, and meets the requirements commonly demanded by labor exchanges, competitive examinations, and professional career evaluation committees.

Title: **Advanced Master's Degree in Mobile Application Development, Android Expert**
Official Nº of Hours: **3,000 h.**


tech technological university

Awards the following

DIPLOMA

to

Mr./Ms. _____, with identification number _____.

For having successfully passed and accredited the following program

ADVANCED MASTER'S DEGREE

in

Mobile Application Development, Android Expert

This is a qualification awarded by this University, equivalent to 3,000 hours, with a start date of dd/mm/yyyy and an end date of dd/mm/yyyy.

TECH is a Private Institution of Higher Education recognized by the Ministry of Public Education as of June 28, 2018.

June 17, 2020

Tere Guevara Navarro
Dean

Unique TECH Code: AFWORD23S techtitute.com/certificates


Advanced Master's Degree in Mobile Application Development, Android Expert

General Structure of the Syllabus

Year	Subject	Hours	Type
1º	Programming Methodologies in Mobile Application Development	150	CO
1º	Technologies in Mobile Application Development	150	CO
1º	Work Tools for Mobile Application Development	150	CO
1º	Multi-Platform Web Development for Mobiles	150	CO
1º	Databases for Mobile Application Development	150	CO
1º	Application Development for iOS Systems	150	CO
1º	Continuous Integration Deployments for Mobiles	150	CO
1º	Mobile User Experience	150	CO
1º	Security on Mobile Devices	150	CO

Year	Subject	Hours	Type
2º	Android Programming Language	150	CO
2º	Frameworks Used in Android Application Development	150	CO
	Interfaces and Layouts in Android Application Development		
2º	Programming Language in Android Applications Kotlin	150	CO
2º	Programming Language in Android Applications	150	CO
2º	Advanced Kotlin Genericity, Functional Programming and Parallelism	175	CO
2º	Data Management on Android Devices	175	CO
2º	Android Device Tools	175	CO
2º	ResponsiveDesign in Android	175	CO
2º	Android Application Marketing	175	CO
2º	Android Application Life Cycle: Cloud, Playstore and Versioning	175	CO

Tere Guevara Navarro
Dean

tech technological university

*Apostille Convention. In the event that the student wishes to have their paper diploma issued with an apostille, TECH EDUCATION will make the necessary arrangements to obtain it, at an additional cost.

tech technological university

Advanced Master's Degree

Mobile Application Development, Android Expert

- Modality: **online**
- Duration: **2 years**
- Certificate: **TECH Technological University**
- Dedication: **16h/week**
- Schedule: **at your own pace**
- Exams: **online**

Advanced Master's Degree

Mobile Application Development, Android Expert

tech technological university