

Master Specialistico

Sviluppo di Applicazioni Mobili,
Esperto in Sistemi Android



Master Specialistico Sviluppo di Applicazioni Mobili, Esperto in Sistemi Android

- » Modalità: **online**
- » Durata: **2 anni**
- » Titolo: **TECH Università
Tecnologica**
- » Orario: **a scelta**
- » Esami: **online**

Accesso al sito web: www.techtute.com/it/informatica/master-specialistico/master-specialistico-sviluppo-applicazioni-mobili-esperto-sistemi-android

Indice

01

Presentazione

pag. 4

02

Obiettivi

pag. 8

03

Competenze

pag. 18

04

Direzione del corso

pag. 22

05

Struttura e contenuti

pag. 30

06

Metodologia

pag. 50

07

Titolo

pag. 58

01

Presentazione

Oggi è inconcepibile un mondo senza applicazioni e aziende che sfruttano le nuove tecnologie. Le applicazioni mobili sono un passo in avanti per qualsiasi azienda, costituiscono un segno di maturità e uno sforzo per adattarsi alle nuove abitudini degli utenti, rendendole essenziali praticamente in ogni campo. Attualmente, la necessità di applicazioni mobili è legata alle funzionalità che possono offrire al di là delle informazioni che può avere una pagina web. Per questo motivo, oggi ci sono più di 3.000 milioni di dispositivi Android nel mondo e, di conseguenza, il settore dello sviluppo di applicazioni mobili ha conosciuto una crescita vertiginosa. Ora più che mai il settore ha bisogno di professionisti specializzati nello Sviluppo di Applicazioni, motivo per cui è stato creato questo Master Specialistico.





“

Osserva, dagli esperti del settore, tutti gli aspetti chiave del Customer Experience per sviluppare applicazioni Android di successo nel mercato”

I rapporti sul mercato del lavoro confermano la crescente domanda di profili esperti nella progettazione di applicazioni mobili, valorizzando l'esperienza completa nel ciclo di vita dello sviluppo, della distribuzione e della monetizzazione delle applicazioni. Il Master Specialistico è finalizzato a fornire all'informatico tutte le conoscenze relative al linguaggio di programmazione delle applicazioni, con particolare attenzione ad Android, alle architetture e alle interfacce utente. Questo programma è stato progettato dai migliori esperti del settore, che costituiscono il personale docente.

In questo modo, l'informatico acquisirà le competenze per comprendere i diversi linguaggi di programmazione per ogni tipo di dispositivo, approfondendo il *responsivedesign*. Ciò consentirà di sviluppare applicazioni adattabili alle diverse caratteristiche di ciascuno dei dispositivi. Allo stesso tempo, il professionista approfondirà settori quali Web Computing, Software, sviluppo aziendale, Marketing e vendite, consulenza, tecnologia mobile multipiattaforma, aziende, mondo automobilistico, domotica, Internet of Things (IoT), banche e droni.

Il programma dispone della migliore metodologia di studio 100% online, che elimina la necessità di frequentare le lezioni in e di rispettare orari fissi e prestabiliti. In questo modo, in 24 mesi gli informatici acquisiranno le basi necessarie per dimensionare la propria attività basata sullo sviluppo di applicazioni o per intraprendere attività di consulenza in tutti gli aspetti legati alla tecnologia mobile su Android.

Questo **Master Specialistico in Sviluppo di Applicazioni Mobili, Esperto in Sistemi Android** possiede il programma più completo e aggiornato del mercato. Le caratteristiche principali del programma sono:

- ◆ Sviluppo di casi di studio pratici presentati da esperti in campo Informatico
- ◆ Contenuti grafici, schematici ed eminentemente pratici che forniscono informazioni scientifiche e pratiche sulle discipline essenziali per l'esercizio della professione
- ◆ Esercizi pratici che offrono un processo di autovalutazione per migliorare l'apprendimento
- ◆ Speciale enfasi sulle metodologie innovative per lo sviluppo di applicazioni mobile
- ◆ Lezioni teoriche, domande all'esperto, forum di discussione su temi controversi e lavoro di riflessione individuale
- ◆ Contenuti disponibili da qualsiasi dispositivo fisso o portatile provvisto di connessione a internet



Con questo Master Specialistico ti aggiornerai e approfondirai i differenti linguaggi di programmazione per ogni tipo di dispositivo mobile”

“

Man mano che padroneggerai tutti i contenuti sullo Sviluppo di Applicazioni Mobili, capirai come monetizzarle e come approfondire il Mobile Marketing”

Il personale docente del programma comprende rinomati professionisti dell'Informatica, che forniscono agli studenti le competenze necessarie a intraprendere un percorso di studio eccellente.

I contenuti multimediali, sviluppati in base alle ultime tecnologie educative, forniranno al professionista un apprendimento coinvolgente e localizzato, ovvero inserito in un contesto reale.

La creazione di questo programma è incentrata sull'Apprendimento Basato su Problemi, mediante il quale lo specialista deve cercare di risolvere le diverse situazioni che gli si presentano durante il corso. Lo studente potrà usufruire di un innovativo sistema di video interattivi creati da esperti di rinomata fama.

Porta soluzioni efficaci all'interno della tua azienda, grazie alla padronanza dei fondamentali in materia di Sviluppo di Applicazioni per Sistemi Android.

TECH ti offre l'opportunità per imparare a sviluppare applicazioni mobili in modo autonomo e professionale, su dispositivi multiplatforma.



02 Obiettivi

A causa della notoria crescita e del consolidamento che porta con sé il mondo delle tecnologie e dei dispositivi mobili, l'informatico è obbligato ad aggiornarsi continuamente per acquisire tutte le conoscenze necessarie. Per questo motivo, questo master specialistico fornisce tutti gli strumenti e le basi necessarie per approfondire lo sviluppo di applicazioni mobile, con particolare attenzione al sistema Android, e fornire soluzioni efficaci in ambito professionale. Il professionista, una volta terminato il Master Specialistico sarà aggiornato sulle ultime tendenze e saprà come operare in tutti gli aspetti, dalla progettazione e programmazione all'esperienza dell'utente finale.





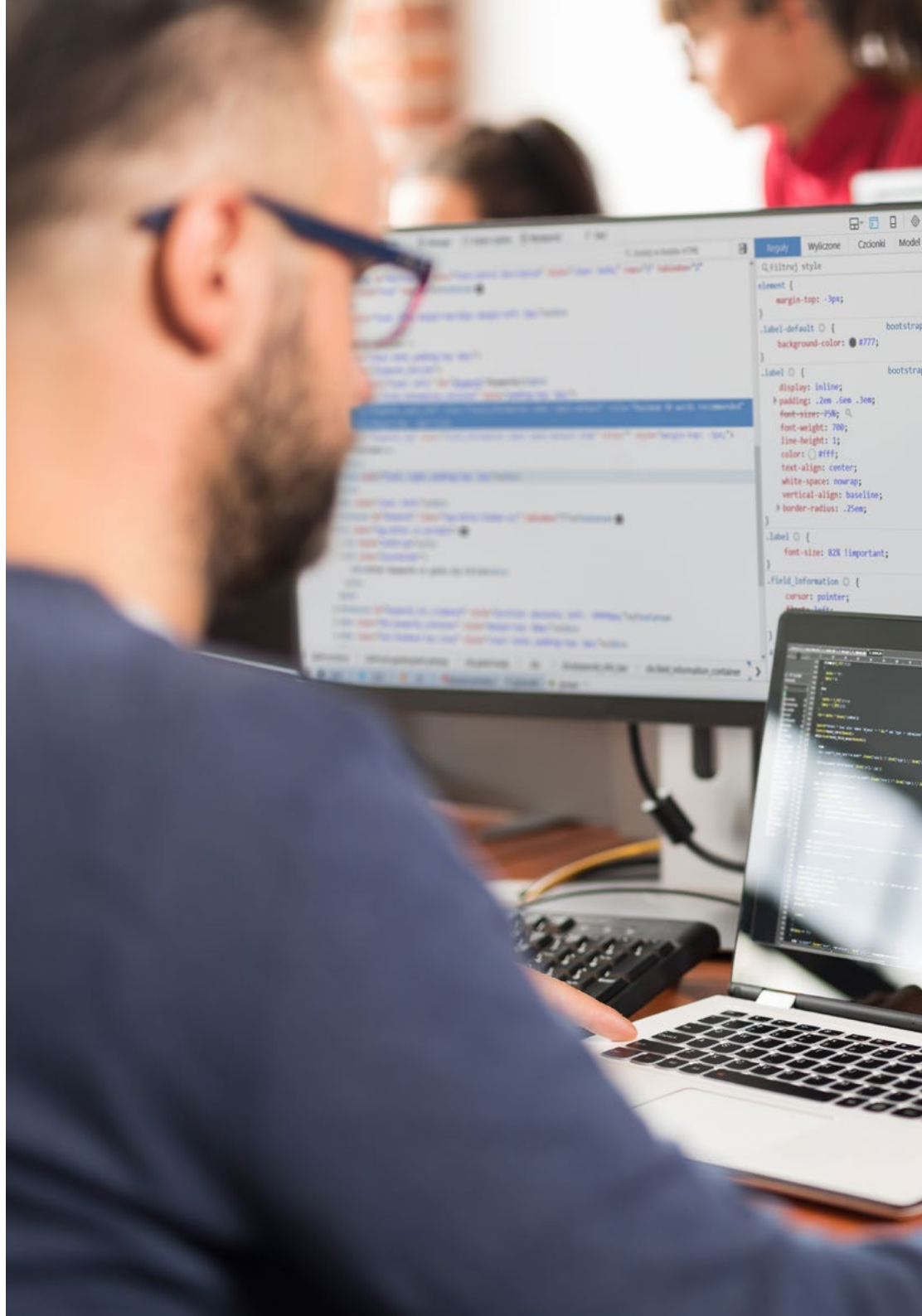
“

Non lasciar correre l'opportunità di prepararti con questo Master Specialistico e riuscire a risolvere efficacemente le problematiche nel contesto professionale”



Obiettivi generali

- ◆ Analizzare le esigenze e il comportamento degli utenti in relazione ai dispositivi mobili e alle loro applicazioni
- ◆ Eseguire la progettazione di architetture, iterazioni e interfacce utente attraverso i linguaggi di programmazione delle piattaforme mobili più rappresentative del mercato (Web, iOS e Android)
- ◆ Applicare meccanismi di controllo degli errori, test e debug nello Sviluppo di Applicazioni mobile
- ◆ Affrontare diversi casi pratici e commerciali per la pubblicazione, la distribuzione e la diffusione di applicazioni mobili nei principali marketplace di applicazioni
- ◆ Acquisire le conoscenze pratiche per pianificare e gestire progetti tecnologici legati alle tecnologie mobili
- ◆ Sviluppare le competenze, le attitudini e gli strumenti necessari per imparare a sviluppare applicazioni mobili in modo autonomo e professionale, su dispositivi multipiattaforma
- ◆ Esplorare i contenuti relativi alla monetizzazione delle applicazioni e al Mobile Marketing







Obiettivi specifici

Modulo 1. Metodologie di Programmazione nello Sviluppo di Applicazioni per Dispositivi Mobili

- ◆ Esplorare processi di sviluppo di un software tradizionale
- ◆ Analisi dei processi di sviluppo agile
- ◆ Promuovere pratiche di sviluppo
- ◆ Esaminare diverse tecniche di rappresentazione e di diagramma
- ◆ Approfondire i diversi modelli di progettazione presenti nell'industria del software
- ◆ Esplorare diverse tecniche di test del software
- ◆ Riconoscere le norme e gli standard di riferimento per la qualità nello sviluppo

Modulo 2. Tecnologie per lo sviluppo di applicazioni per dispositivi mobili

- ◆ Stabilire i concetti per i dispositivi mobili
- ◆ Compilazione delle piattaforme principali
- ◆ Esaminare i loro componenti comuni
- ◆ Identificare i componenti differenzianti, le loro capacità e i loro limiti
- ◆ Delimitare i diversi scenari in cui possono operare Vantaggi
- ◆ Analizzare le diverse interazioni che questi dispositivi possono mediare
- ◆ Sensibilizzare sui diversi abusi che possono essere commessi

Modulo 3. Strumenti del Lavoro nello Sviluppo di Applicazioni per Dispositivi Mobili

- ◆ Preparazione dell'ambiente di Sviluppo
- ◆ Acquisire competenze sul terminale di comando
- ◆ Utilizzare in modo efficiente il sistema di controllo delle versioni
- ◆ Affrontare l'uso di sistemi di versionamento del codice remoto
- ◆ Stabilire le nozioni fondamentali sul funzionamento di Internet
- ◆ Sviluppare i concetti rilevanti della programmazione software
- ◆ Esaminare le strutture di dati
- ◆ Esaminare le tecniche di progettazione e interpretazione degli algoritmi

Modulo 4. Sviluppo web multiplatforma per dispositivi mobili

- ◆ Determinare i vantaggi e i limiti del modello di sviluppo nativo e dello Sviluppo di Apps ibride
- ◆ Esaminare le caratteristiche e i limiti delle Progressive Web Apps (PWA)
- ◆ Analizzare i principali *Framework* per lo sviluppo di applicazioni Web: Angular, React, Vue
- ◆ Compilare le principali tecnologie per lo sviluppo di applicazioni mobili multiplatforma: *Ionic* e *Flutter*
- ◆ Analizzare le capacità di distribuire queste applicazioni ibride come applicazioni web o desktop sui PC
- ◆ Esaminare un modello per scegliere l'alternativa più adatta allo sviluppo di un'applicazione specifica

Modulo 5. Database per lo sviluppo di applicazioni per dispositivi mobili

- ◆ Identificare il modello di database migliore in relazione alle caratteristiche dell'applicazione mobile
- ◆ Stabilire le capacità di ciascun sistema di database
- ◆ Determinare le differenze tra i vari database
- ◆ Esaminare come connettersi e caricare/estrarre dati da diversi tipi di database
- ◆ Analizzare le funzionalità di base degli ambienti di sviluppo, comprese quelle del database

Modulo 6. Sviluppo di applicazioni per sistemi iOS

- ◆ Sviluppare un'applicazione in Swift
- ◆ Usare Cocoa Pods per gestire le librerie
- ◆ Fare uso di Alamofire per collegare la nostra applicazione a un'API RESTful
- ◆ Specificare i requisiti di base per la scelta di una libreria
- ◆ Sviluppare un sistema di monetizzazione con ADMOB
- ◆ Progettare le viste dal codice
- ◆ Pubblicare un'applicazione su App Store

Modulo 7. Distribuzioni a integrazione continua per dispositivi mobili

- ◆ Determinare lo scenario peggiore che dà origine alla necessità di questa metodologia
- ◆ Specificare i requisiti che il Software da integrare deve soddisfare
- ◆ Stabilire cosa si intende per integrazione continua, consegna continua e distribuzione continua
- ◆ Analizzare DevSecOps
- ◆ Esaminare il monitoraggio in corso
- ◆ Sviluppare le implementazioni delle diverse fasi

Modulo 8. User Experience su dispositivi mobili

- ◆ Analizzare il nuovo tipo di utente, le sue interazioni e il suo percorso attraverso le applicazioni mobili e i siti web
- ◆ Determinare gli strumenti fondamentali per la web analytics, la mobilità e l'accessibilità
- ◆ Concretizzare le tecniche di valutazione delle micro-interazioni e la progettazione di esperienze personalizzate
- ◆ Stabilire come le nuove tecnologie dirompenti come l'AI e l'IoT abbiano portato l'esperienza del cliente a nuovi standard
- ◆ Mostrare come l'analisi comportamentale generi una quantità e una qualità di dati sconosciuta all'analitica tradizionale
- ◆ Sviluppare nuove metodologie, come il *Design Thinking*, incentrate sull'utente
- ◆ Proporre strumenti di prototipazione e *wireframing* di base e avanzati.

Modulo 9. Sicurezza sui dispositivi mobili

- ◆ Determinare le caratteristiche e i livelli di sicurezza di un dispositivo mobile
- ◆ Stabilire le tecniche da utilizzare per un dispositivo
- ◆ Analizzare gli errori di sicurezza più comuni
- ◆ Esaminare i meccanismi abilitati nella programmazione per evitare violazioni della sicurezza
- ◆ Concretizzare le raccomandazioni fornite dalle agenzie di sicurezza
- ◆ Compilare le diverse soluzioni disponibili sul mercato per la gestione della sicurezza dei dispositivi mobili all'interno delle aziende
- ◆ Analizzare i processi crittografici applicabili alla sicurezza mobile

Modulo 10. Linguaggio di programmazione Android

- ◆ Esaminare il nucleo Linux e la macchina virtuale in Android
- ◆ Analizzare la libreria di sistema nativa
- ◆ Stabilire i vantaggi di Android rispetto ad altre piattaforme
- ◆ Determinare gli elementi di un'applicazione Android
- ◆ Presentare le versioni Android e i suoi aggiornamenti
- ◆ Valutare il mercato di applicazioni Android
- ◆ Prevedere l'evoluzione futura di Android

Modulo 11. Frameworks usati nello sviluppo di applicazioni Android

- ◆ Analizzare il *framework* di Android Core
- ◆ Sviluppare altri *frameworks* usati per lo sviluppo di applicazioni in Android
- ◆ Implementare librerie con *gradle*
- ◆ Stabilire i *frameworks* per connettere una API
- ◆ Generare Conoscenze specialistiche sulla tecnologia MVP
- ◆ Connettere pro e contro su MVP y MVVM

Modulo 12. Interfacce e layouts nello sviluppo di applicazioni Android

- ◆ Presentare il modello del ciclo di vita delle visite su Android
- ◆ Esaminare gli attributi più importanti di un design visivo (*layout*)
- ◆ Analizzare i design (*layout*) disponibili
- ◆ Generare un design (*layout*) riutilizzabile
- ◆ Determinare l'utilizzo di risorse alternative
- ◆ Identificare le differenze di uso di queste componenti confrontandoli con altri sistemi di programmazione
- ◆ Stabilire la potenzialità e uso del file *AndroidManifest.xml*

Modulo 13. Linguaggio di programmazione in applicazioni Android. Kotlin

- ◆ Sviluppare il linguaggio di programmazione Kotlin
- ◆ Compilare le caratteristiche e capacità differenziali del Linguaggio Kotlin
- ◆ Esaminare il modello di esecuzione di base di un programma in Kotlin
- ◆ Analizzare la sintassi del linguaggio e la struttura di un programma in Kotlin
- ◆ Connettere il modello di tipi di variabili in Kotlin
- ◆ Stabilire le diverse forme di gestione del flusso del codice in Kotlin
- ◆ Determinare il modello di classe, collezione e oggetti in Kotlin
- ◆ Generare conoscenze specialistiche sul modello ereditario in Kotlin
- ◆ Connettere il modello di gestione di eccezioni e tipi null in Kotlin

Modulo 14. Linguaggio di programmazione in applicazioni Android. Kotlin avanzato. Generalità, programmazione funzionale e parallelismo

- ◆ Esaminare il modello di Generalità Covariante e Contravariante in Kotlin
- ◆ Analizzare il modello di programmazione funzionale con *Lambdas* in Kotlin
- ◆ Connettere le funzioni di ordine superiore in Kotlin
- ◆ Sviluppare le estensioni e gli oggetti compagni in Kotlin
- ◆ Esaminare l'uso del *Delegation* pattern in Kotlin
- ◆ Compilare le annotazioni e la riflessione in Kotlin
- ◆ Approfondire il modello di *Testing* in Kotlin
- ◆ Stabilire i diversi modelli di programmazione asincrona disponibili in Kotlin
- ◆ Determinare il modello di corruttori di Kotlin
- ◆ Compilare le diverse librerie e strumenti di utilità dell'ecosistema Kotlin

Modulo 15. Gestione dei dati sui dispositivi Android

- ◆ Analizzare le diverse tecniche per la gestione dei dati in Android
- ◆ Proporre metodi d'uso ottimale di dati sul dispositivo
- ◆ Identificare gli strumenti necessari per la ottimizzazione di I dati
- ◆ Esaminare le caratteristiche di JSON e XML per la gestione dei dati in Android
- ◆ Valutare le problematiche generali di Sistemi Distribuiti applicabili al mondo delle applicazioni dei dispositivi mobili
- ◆ Determinare l'uso della libreria *Room* come astrazione per l'uso di SQLite in Android e i suoi vantaggi e svantaggi
- ◆ Stabilire i permessi di sicurezza necessari per la Gestione dei Dati con qualsiasi tecnica usata in Android

Modulo 16. Strumenti dei Dispositivi Android

- ◆ Conoscere gli strumenti più avanzati nella gestione quotidiana
- ◆ Valutare gli strumenti di controllo dei dispositivi Android
- ◆ Dimostrare i vantaggi dell'utilizzo di Android sui droni
- ◆ Specificare le funzionalità degli strumenti CRM
- ◆ Dimostrare i vantaggi dei dispositivi Android nelle piattaforme Bancarie
- ◆ Analizzare le possibilità dell'IoT e delle piattaforme Android
- ◆ Esaminare gli strumenti di efficienza dei processi su Android

Modulo 17. Design *responsive* su Android

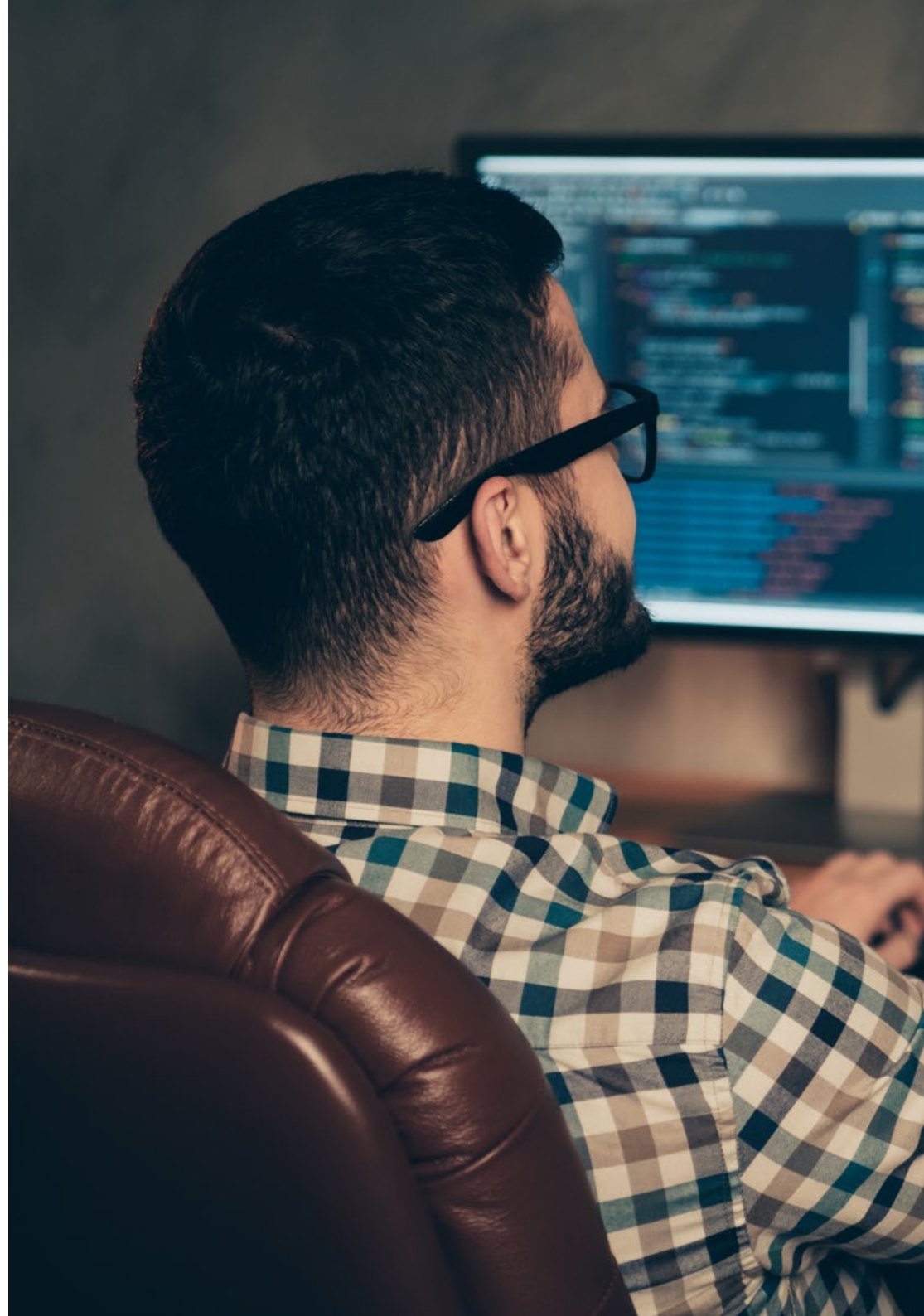
- ◆ Analizzare gli elementi principali di un design
- ◆ Definire una metodologia di design di elementi visivi e una metodologia screen design
- ◆ Risolvere i diversi problemi che si affrontano nello sviluppo multidispositivo
- ◆ Disporre di strumenti per generare più risorse e migliori per lo sviluppo multidispositivo
- ◆ Esaminare *frameworks* alternativi allo sviluppo response nativo
- ◆ Generare una metodologia propria per sviluppare applicazioni usando buone prassi per la visualizzazione su più dispositivi dall'inizio del progetto

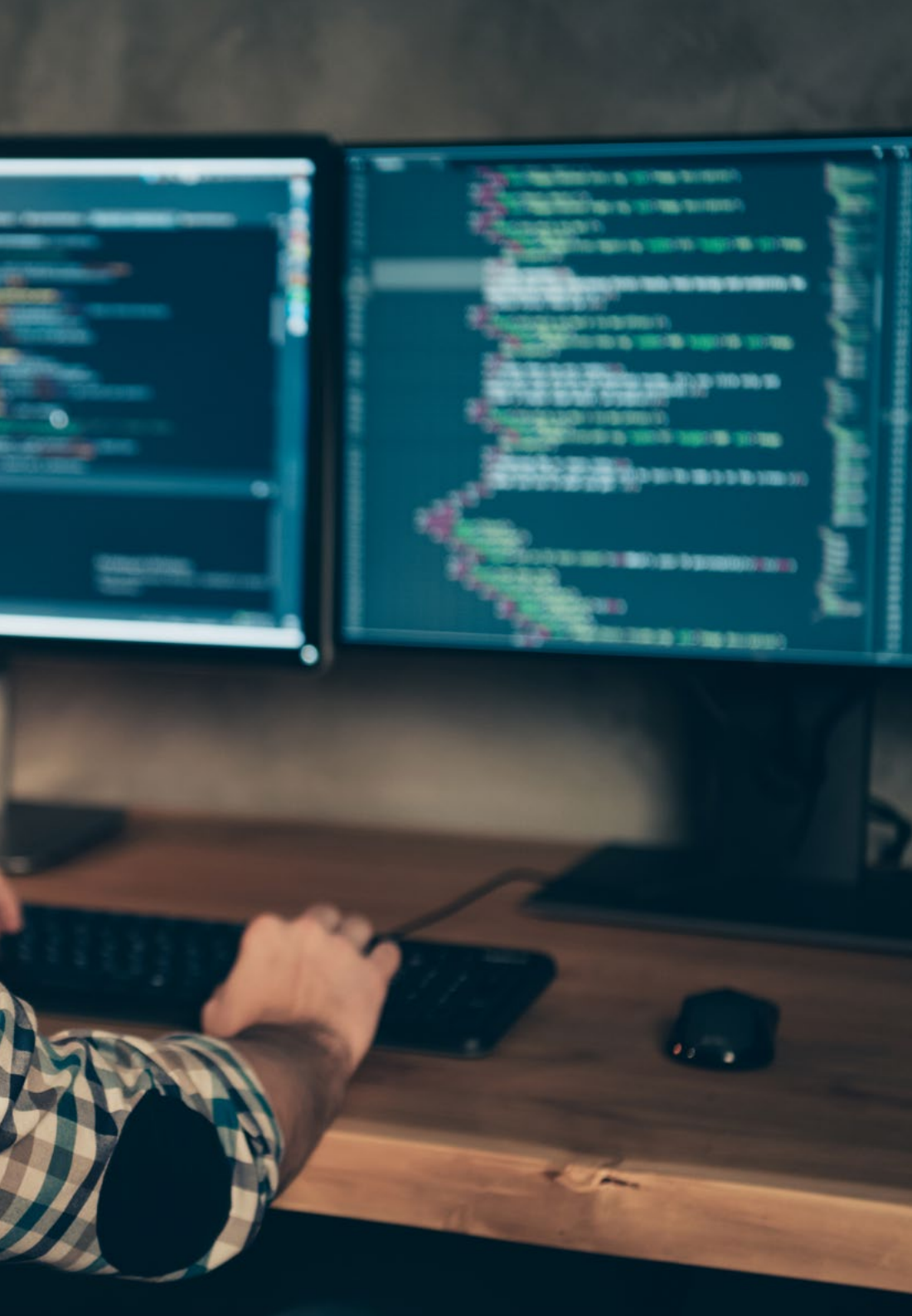
Modulo 18. Marketing per applicazioni Android

- ◆ Analizzare nuove metodologie incentrati sull'utente
- ◆ Determinare come l'Intelligenza Artificiale abbia portato la CX al livello successivo
- ◆ Stabilire l'importanza dell'accessibilità e mobilità
- ◆ Sviluppare tecniche d'analisi più avanzate di sessione e comportamento
- ◆ Concretizzare obiettivi di micropersonalizzazione nel *journey* dell'utente
- ◆ Compilare le nuove metodologie proprie di un ambiente vivo e dinamico
- ◆ Proporre tecniche per la creazione di prototipi

Modulo 19. Ciclo di vita in applicazioni Android. Cloud, playstore e versioni

- ◆ Concretizzare i benefici da adottare in un modello automatizzato di distribuzione delle versioni
- ◆ Stabilire le differenze tra integrazione continua, consegna e aggiornamenti continui
- ◆ Definire le caratteristiche principali DevOps
- ◆ Valutare alcuni strumenti fondamentali per attuare pipelines di CI/CD
- ◆ Concretizzare i fattori essenziali per sviluppare applicazioni preparate a sopportare processi CI/CD
- ◆ Esaminare le Tecnologie dei Contenitori come pilastri fondamentali del CI/CD
- ◆ Identificare pratiche, casistica di uso, tecnologie e strumenti dell'ecosistema CI/CD





“

Acquisirai conoscenze specializzate sul ciclo di vita delle applicazioni Android con il fine di lavorare in modo più efficiente come sviluppatore”

03

Competenze

L'industria tecnologica richiede costantemente profili professionali multitasking, il che implica un aggiornamento costante per operare in un mercato in vertiginosa evoluzione. Questo Master Specialistico prepara informatici con conoscenze specialistiche, aggiornate sulle diverse componenti tecnologiche dei dispositivi mobili per lo sviluppo di applicazioni. Il professionista avrà una conoscenza approfondita dei migliori *Hardwar* e *Softwar* indispensabili per ogni professionista che opera in questo settore. In questo modo, l'informatico sarà aggiornato con le conoscenze necessarie per essere autonomo nello sviluppo di un'applicazione per dispositivi Android, dalla progettazione alla produzione.



“

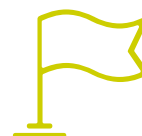
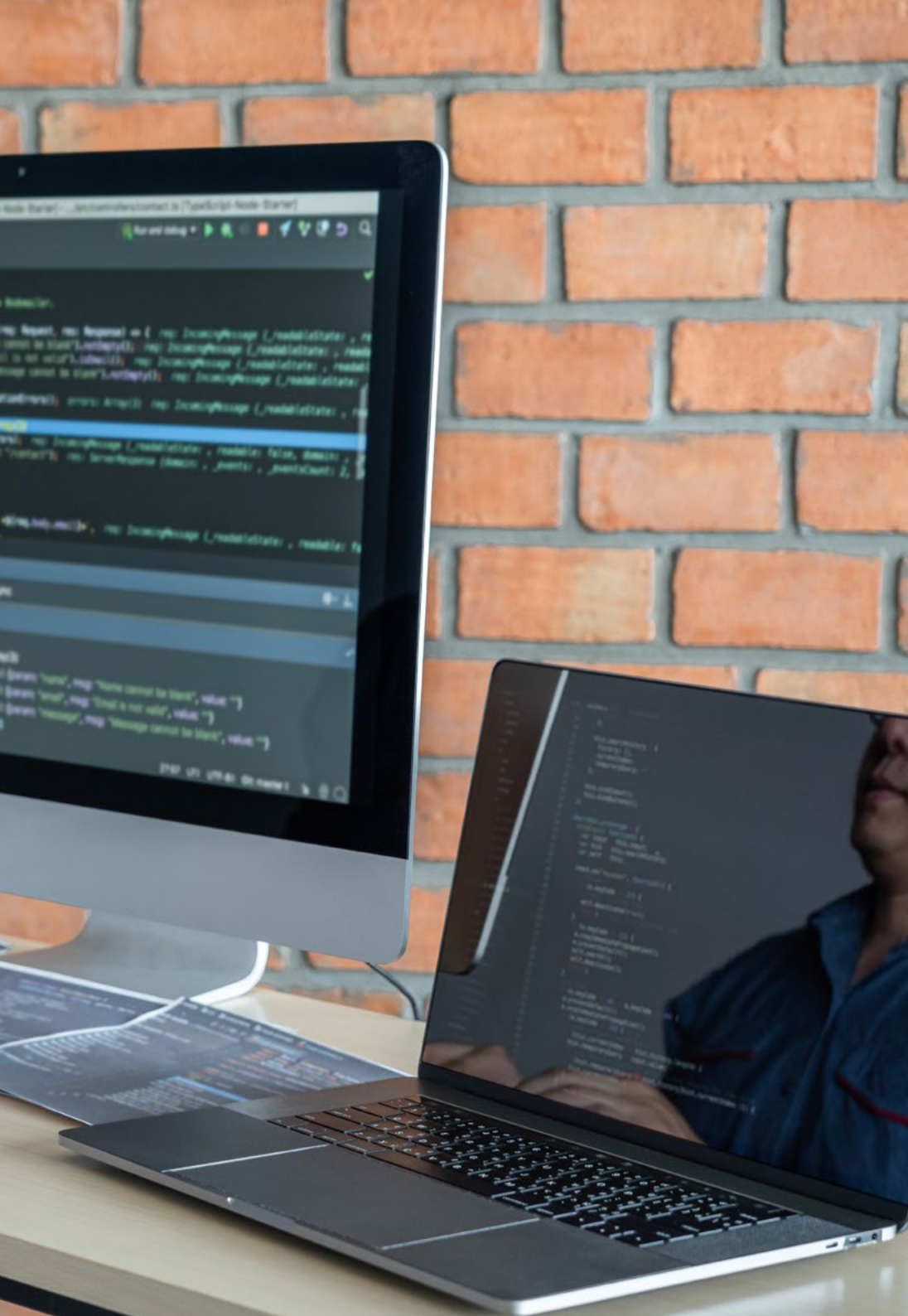
Grazie a questo Master Specialistico darai una svolta alla tua carriera grazie alle conoscenze pratiche per pianificare e dirigere progetti con tecnologie mobili”



Competenze generali

- ◆ Sviluppare applicazioni mobili sia per dispositivi Android che iOS
- ◆ Potenziare le *skills* necessarie in tutti i punti del ciclo di vita dello Sviluppo del Software
- ◆ Analizzare le diverse metodologie di Programmazione per i differenti dispositivi e casistiche d'uso
- ◆ Padroneggiare conoscenze specializzate in differenti componenti tecnologiche per i dispositivi mobili
- ◆ Comprendere l'ambiente di lavoro per i diversi linguaggi e stili di programmazione di applicazioni mobili
- ◆ Realizzare efficacemente le applicazioni dal punto di vista dell'esperienza dell'utente in modo che sia realmente funzionale e attrattiva
- ◆ Padroneggiare gli elementi strutturali di un Sistema Android
- ◆ Analizzare i diversi *Frameworks* per utilizzarli su Architetture multiple
- ◆ Definire le proprietà di un'interfaccia dal punto di vista dell'usabilità
- ◆ Sviluppare conoscenze specializzate sul linguaggio di programmazione Kotlin e sul contesto di apparizione
- ◆ Gestire il *Content Provider* per lo scambio di dati e la sua sicurezza
- ◆ Conoscere gli strumenti aziendali su Android: efficienza e utilità
- ◆ Analizzare gli elementi un responsive design
- ◆ Esplorare nuove metodologie incentrati sull'utente
- ◆ Adattare il Sistema di Pipelines Automatizzati all'Ecosistema Android





Competenze specifiche

- ◆ Analizzare i Processi di Sviluppo di un Software dal punto di vista tradizionale e smart
- ◆ Sviluppare le considerazioni generali sui dispositivi mobili
- ◆ Padroneggiare i concetti chiave della Programmazione e Internet, il Web e il suo funzionamento
- ◆ Eseguire il confronto tra il modello di sviluppo nativo e il modello di sviluppo web multiplatforma basato su app ibride
- ◆ Scelta del database in applicazioni mobili
- ◆ Pubblicare un'applicazione su *Play Store*
- ◆ Determinare le diverse tappe di un ciclo di integrazione continua
- ◆ Comprendere i principi della cultura centrata sull'utente e il modo in cui essa genera una nuova posizione per i professionisti della customer experience
- ◆ Saper risolvere i problemi di sicurezza esistenti nei dispositivi mobili
- ◆ Utilizzare risorse alternative analizzando i design (*layout*) disponibili
- ◆ Sviluppare le diverse forme di gestione del flusso del codice in Kotlin e le sue capacità
- ◆ Migliorare la produttività delle applicazioni attraverso le capacità differenziali del Linguaggio Kotlin
- ◆ Sviluppare le estensioni e gli oggetti compagni in Kotlin
- ◆ Usare i database o servizi online per creare file in differenti formati
- ◆ Determinare l'uso della libreria *Room* come astrazione per l'uso di SQLite in Android
- ◆ Valutare gli strumenti di controllo dei dispositivi Android, analizzando le possibilità dell'IoT e delle piattaforme Android
- ◆ Adottare il *responsive design* per dare esperienze d'utilizzo più soddisfacenti
- ◆ Combinare le Tecniche di Prototipazione e *Wireframing* con le nuove tecnologie dirompenti come l'Intelligenza Artificiale e the Internet of Things

04

Direzione del corso

Questo Master Specialistico in Sviluppo di Applicazioni Mobili, Esperto in Sistemi Android è guidato da un personale docente con una vasta conoscenza ed esperienza nelle nuove tecnologie, nell'architettura delle soluzioni e nell'infrastruttura digitale, da esperti di programmazione Android e da sviluppatori di applicazioni. In questo modo, si offre all'informativo una garanzia di qualità del contenuto scelto per questo Master Specialistico. Scommettendo per l'ottimizzazione dei processi di apprendimento degli studenti che cercano in questo modo il contributo necessario per il loro successo professionale.



“

Godere di tutte le esperienze di informatici e ingegneri esperti in Sviluppo di Applicazioni, aggiungerà valore alle tue proposte e ti renderà più interessante all'interno del mercato del lavoro”

Direttore Ospite Internazionale

Colin Lee è uno sviluppatore di applicazioni mobili di successo, specializzato in codice nativo Android, la cui influenza si estende a livello internazionale. L'esperto è un'autorità nell'area delle Twin Cities e nella gestione di Kotlin. Uno dei suoi contributi più recenti è stato quello di dimostrare, con codice live, come costruire rapidamente un browser utilizzando il suddetto linguaggio di programmazione e i componenti open source del browser di Mozilla per Android.

Inoltre, le sue applicazioni sono state collegate a importanti aziende globali. Ad esempio, è stato responsabile della creazione di soluzioni digitali per Pearson, uno dei maggiori editori al mondo. Ha anche sviluppato un registratore video Android di basso livello per la startup Flipgrid, in seguito acquisita da Microsoft.

Ha anche costruito una VPN Android di successo per un grande cliente di consulenza. È anche il creatore di uno strumento di gestione delle merci implementato dalla transnazionale Amazon per facilitare il lavoro dei suoi camionisti a contratto. Ha anche contribuito a costruire le versioni mobili del browser Firefox per Mozilla.

Oggi lavora come appaltatore, tra cui revisioni del codice e controlli di sicurezza. Il suo impatto sullo sviluppo di applicazioni mobili e la sua esperienza nel corso degli anni lo rendono una figura di spicco nell'arena tecnologica globale.



Dott. Lee, Colin

- Direttore presso ColinTheShots LLC
- Ingegnere software Android presso Specto Inc.
- Ingegnere Senior Android presso Mozilla
- Ingegnere Sviluppatore Software presso Amazon
- Ingegnere di Applicazioni Mobili presso Flipgrid
- Specialista di Configurazione Software per Pearson VUE
- Laurea presso l'Università della Florida

“

Grazie a TECH potrai imparare con i migliori professionisti del mondo”

Direzione



Dott. Olalla Bonal, Martín

- *Blockchain Technical Specialist* in IBM SPGI
- Tecnico di Elettronica Digitale
- Architetto *Blockchain*
- Architetto di Infrastrutture nel Settore Bancario
- Formazione *Hyperledger Fabric* per le aziende
- Formazione sulla *Blockchain* orientata al business nelle aziende
- Gestione del progetto e implementazione delle soluzioni
- Oltre 25 anni di esperienza nel Settore IT

Personale docente

Dott. Gozalo Fernández, Juan Luis

- ◆ Ingegnere informatico
- ◆ Professore associato in Programmi Universitari in DevOps e Sviluppo di Applicazioni *Blockchain* presso l'Universidad Internacional de La Rioja
- ◆ Ex-direttore Blockchain DevOps presso Alastria
- ◆ Direttore della Tecnologia a Livello di Servizio presso Santander Spagna
- ◆ Responsabile per lo Sviluppo dell'Applicazione Mobile di Tinkerlink presso Cronos Telecom
- ◆ Direttore della Tecnologia di Gestione dei Servizi IT presso Barclays Bank Spagna
- ◆ Laurea in Ingegneria Informatica presso l'UNED
- ◆ Specializzazione in Deep Learning presso DeepLearning.ai

Dott. Gómez Rodríguez, Antonio

- ◆ Ingegnere delle soluzioni Cloud in Oracle
- ◆ Direttore di Progetti in importanti gruppi imprenditoriali come Sopra Group ed Everis
- ◆ Project Manager presso la Società pubblica per la Gestione di Programmi Culturali del Consiglio della Cultura dell'Andalusia
- ◆ Laurea in Ingegneria delle Telecomunicazioni presso l'Università Politecnica della Catalogna
- ◆ Specializzazione in Tecnologie e Sistemi Informatici presso l'Istituto Catalano di Tecnologia
- ◆ Master E-Business presso la Business School di La Salle
- ◆ Analista di Sistemi Informatici. Sopra Group

Dott. Guerrero Díaz-Pintado, Arturo

- ◆ Ingegnere di rete R&D presso Telefónica
- ◆ Ingegnere Tecnico di Prevendita attraverso *Watson Customer Engagement* portfolio (soluzioni di Marketing e *Customer Experience*) in Spagna, Portogallo, Grecia e Israele in IBM
- ◆ Consulente di servizi professionali lavorando con organizzazioni all'avanguardia in Europa, Medio Oriente e America Latina in IBM
- ◆ Laurea in Ingegneria delle Telecomunicazioni presso l'Università di Alcalá e la *Danish Technical University*
- ◆ Ha realizzato importanti collaborazioni in riconosciute università e centri d'educazione superiore in materie relate con la tecnologia come Intelligenza Artificiale, *Internet of Things*, *Cloud*, *Customer Experience* e *Digital Transformation*

Dott. Villot Guisán, Pablo

- ◆ Architetto Cloud di Soluzioni Esponenziali e Subject Mater Expert *Blockchain* in KPMG
- ◆ Architetto Cloud di Soluzioni Esponenziali e Subject Mater Expert *Blockchain* di Integrazione in Everis
- ◆ Sviluppatore e Responsabile Tecnico di applicazioni web e di scrittura per l'area di Logistica Commerciale di Inditex, Connectis
- ◆ Certificato Microsoft presso MSCA: *Cloud Platform*
- ◆ Laureato in Ingegneria Informatica presso l'Università della Coruña

Dott. Pérez Rico, Javier

- ◆ Responsabile tecnico Android presso Nologis
- ◆ Responsabile tecnico Android presso Seekle
- ◆ Programmatore Android presso Gowex-Ideup
- ◆ Programmatore Junior Android presso Tecnom
- ◆ Relatore al II Simposio iTest, E@tic2011
- ◆ Laurea in Ingegneria Tecnico di Sistemi Informatici presso l'Università Complutense di Madrid
- ◆ Master in Ricerca presso l'Università Complutense di Madrid

Dott. Noguera Rodríguez, Pablo

- ◆ Sviluppatore Java esperto: JSE, JEE y Android - Ilabora Formación
- ◆ Programmazione di applicazioni Android - EOI - Madrid
- ◆ Sviluppatore di app native (iOS e Android) - Starman Aviation (App Aviaze)
- ◆ Sviluppatore di app native (iOS) - Stef (App Mtrack)
- ◆ Sviluppatore di app native (iOS e Android) - Bitnovo (App Bitnovo)





Dott. Frias Favero, Pedro Luis

- ◆ CTO-Swearit Technologies
- ◆ COO - Key Identification
- ◆ Full Stack Developer - Ironhack
- ◆ Laurea in Ingegneria Industriale presso l'Università di Yacambú
- ◆ Esperto in *Blockchain* e Applicazioni Decentrate dell'Università di Alcalá

Dott. Jiménez Pérez, Carlos

- ◆ Sviluppatore Android Senior in OnThe Spot - Telefónica Tech
- ◆ Professore Strutturato presso Università Carlos III di Madrid
- ◆ Laurea in Ingegneria in Automatica ed Elettronica
- ◆ Master in Ingegneria dei Sistemi Elettronici e Applicazioni

05

Struttura e contenuti

Con l'obiettivo di fornire all'informatico i migliori contenuti per lo sviluppo di applicazioni mobili di successo, TECH si avvale di un team di professionisti della programmazione, dello sviluppo di applicazioni e delle nuove tecnologie che hanno selezionato contenuti esclusivi e che lo accompagneranno nel processo di studio durante i moduli da sviluppare in 24 mesi. Da una piattaforma online, il professionista potrà accedere a contenuti interattivi, distribuiti in video, grafici, esempi basati su problemi, materiale audiovisivo e con una vasta comunità per discutere le questioni sollevate.



“

Avrai a disposizione i migliori contenuti audiovisivi con esercizi pratici e formati interattivi per rendere il tuo processo di apprendimento agile ed efficiente"

Modulo 1. Metodologie di Programmazione nello Sviluppo di Applicazioni per Dispositivi Mobili

- 1.1. Processi di Sviluppo di un Software
 - 1.1.1. *Waterfall*
 - 1.1.2. *Spiral*
 - 1.1.3. RUP
 - 1.1.4. V-Model
- 1.2. Processi di Sviluppo di Software Agili
 - 1.2.1. *Scrum*
 - 1.2.2. XP
 - 1.2.3. Kanban
- 1.3. Linguaggio di Modellazione Unificato (UML)
 - 1.3.1. UML
 - 1.3.2. Tipi di modellamento
 - 1.3.3. Blocchi di base di UML
- 1.4. Diagrammi Comportamentali UML
 - 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. Diagrammi Strutturali UML
 - 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. Modelli di design creativi
 - 1.6.1. *Singleton*
 - 1.6.2. *Prototype*
 - 1.6.3. *Builder*
 - 1.6.4. *Factory*
 - 1.6.5. *Abstract factory*
- 1.7. Modelli di Progettazione strutturale
 - 1.7.1. *Decoratore*
 - 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. Modelli di comportamento
 - 1.8.1. *Chain of responsibility*
 - 1.8.2. *Command*
 - 1.8.3. *Interlocutore*
 - 1.8.4. *Mediatore*
 - 1.8.5. *Mediatore*
 - 1.8.6. *Observer*
 - 1.8.7. *State*
 - 1.8.8. *Strategy*
 - 1.8.9. *Template Method*
 - 1.8.10. *Visitatore*
- 1.9. *Testing*
 - 1.9.1. Test unitari
 - 1.9.2. Test di integrazione
 - 1.9.3. Tecniche di scatola bianca
 - 1.9.4. Tecniche di scatola nera
- 1.10. Qualità
 - 1.10.1. ISO
 - 1.10.2. ITIL
 - 1.10.3. COBIT
 - 1.10.4. PMP

Modulo 2. Tecnologie per lo sviluppo di applicazioni per dispositivi mobili

- 2.1. Dispositivi mobili
 - 2.1.1. Dispositivi mobili
 - 2.1.2. Infrastruttura di un dispositivo mobile
 - 2.1.3. Produttori di hardware
 - 2.1.4. Sviluppatori di software
 - 2.1.5. Fornitori di servizi
 - 2.1.6. Fornitori di piattaforme
 - 2.1.7. Piattaforme principali
- 2.2. Componenti fisici dei dispositivi mobili
 - 2.2.1. Conservazione
 - 2.2.1.1. Immutabile
 - 2.2.1.2. Mutevole
 - 2.2.1.3. Temporanea
 - 2.2.1.4. Esterna
 - 2.2.2. Presentatori
 - 2.2.2.1. Display, altoparlanti, risposte aptiche
 - 2.2.3. Metodi di ingresso
 - 2.2.3.1. Pulsanti/tastiere
 - 2.2.3.2. Schermi
 - 2.2.3.3. Microfoni
 - 2.2.3.4. Sensori di movimento
 - 2.2.4. Fonti di energia
 - 2.2.4.1. Fonti di energia
 - 2.2.4.2. Uso adattativo delle risorse
 - 2.2.4.3. Programmazione efficiente
 - 2.2.4.4. Sviluppo sostenibile
- 2.3. Processori
 - 2.3.1. Processore centrale
 - 2.3.2. Altri processori astratti
 - 2.3.3. Processori di intelligenza artificiale



- 2.4. Trasmettitori di informazioni
 - 2.4.1. Lungo raggio
 - 2.4.2. Medio raggio
 - 2.4.3. Corto raggio
 - 2.4.4. Campo ultra corto
- 2.5. Sensori
 - 2.5.1. Interno al dispositivo
 - 2.5.2. Ambientali
 - 2.5.3. Medici
- 2.6. Componenti logici
 - 2.6.1. Immutabili
 - 2.6.2. Produttore mutabile
 - 2.6.3. A disposizione dell'utente
- 2.7. Categorizzazione
 - 2.7.1. Portatili
 - 2.7.2. Smartphone
 - 2.7.2.1. Tablet
 - 2.7.2.2. Dispositivi multimediali
 - 2.7.2.3. Componenti aggiuntivi intelligenti
 - 2.7.3. Assistenti robotici
- 2.8. Modalità di funzionamento
 - 2.8.1. Offline
 - 2.8.2. Online
 - 2.8.3. Sempre disponibile
 - 2.8.4. Da punto a punto
- 2.9. Interazioni
 - 2.9.1. Interazioni mediate dall'utente
 - 2.9.2. Interazioni mediate dal fornitore
 - 2.9.3. Interazioni mediate da dispositivi
 - 2.9.4. Interazioni mediate dall'ambiente

- 2.10. Sicurezza
 - 2.10.1. Misure attuate dal produttore
 - 2.10.2. Misure attuate dai fornitori
 - 2.10.3. Sicurezza implementata dall'utente
 - 2.10.4. La privacy

Modulo 3. Strumenti del Lavoro nello Sviluppo di Applicazioni per Dispositivi Mobili

- 3.1. Ambiente e strumenti per lo Sviluppo di Applicazioni per dispositivi mobili
 - 3.1.1. Preparazione dell'Ambiente per Mac OS
 - 3.1.2. Preparazione dell'Ambiente per Linux
 - 3.1.3. Preparazione dell'Ambiente per Windows
- 3.2. Linea di Comando
 - 3.2.1. Linea di Comando
 - 3.2.2. Emulatori
 - 3.2.3. Interprete dei comandi
 - 3.2.4. Creazione di cartelle
 - 3.2.5. Creazione di file
 - 3.2.6. Navigazione
 - 3.2.7. Gestione di file e cartelle tramite l'interfaccia della riga di comando
 - 3.2.8. Permessi
 - 3.2.9. SSH
 - 3.2.10. Lista di comandi
- 3.3. Deposito Software. Git
 - 3.3.1. Sistemi di controllo delle versioni
 - 3.3.2. Git
 - 3.3.3. Configurazione
 - 3.3.4. Repository
 - 3.3.5. Rami
 - 3.3.6. Gestione delle filiali
 - 3.3.7. Flussi di lavoro
 - 3.3.8. Merge
 - 3.3.9. Comandi

- 3.4. Servizio Web di controllo delle versioni
 - 3.4.1. Depositi remoti
 - 3.4.2. Configurazione
 - 3.4.3. Autenticazione
 - 3.4.4. Forcella del Software
 - 3.4.5. Comando Git Clone
 - 3.4.6. Depositi
 - 3.4.7. *Github Pages*
- 3.5. Strumenti Avanzati di Sviluppo per Applicazioni sui dispositivi mobili
 - 3.5.1. *Postman*
 - 3.5.2. *Visual studio code*
 - 3.5.3. GUI per i database
 - 3.5.4. *Hosting*
 - 3.5.5. Strumenti complementari per lo Sviluppo
- 3.6. Web dal punto di vista dello Sviluppo di Applicazioni mobili
 - 3.6.2. Protocolli
 - 3.6.3. Fornitore di servizi Internet
 - 3.6.4. Indirizzi IP
 - 3.6.5. Servizi di Nomi DNS
- 3.7. Programmazione nello Sviluppo di Applicazioni per dispositivi mobili
 - 3.7.1. Programmazione nello Sviluppo di Applicazioni per dispositivi mobili
 - 3.7.2. Paradigmi della programmazione
 - 3.7.3. Linguaggi di programmazione
- 3.8. Componenti di Sviluppo di Applicazioni per dispositivi mobili
 - 3.8.1. Variabili e costanti
 - 3.8.2. Tipologie
 - 3.8.3. Operatori
 - 3.8.4. Dichiarazioni
 - 3.8.5. Loop
 - 3.8.6. Funzioni e oggetti

- 3.9. Struttura dei dati
 - 3.9.1. Struttura dei dati
 - 3.9.2. Tipi di struttura lineari
 - 3.9.3. Tipi di struttura funzionale
 - 3.9.4. Tipi di strutture ad albero
- 3.10. Algoritmi
 - 3.10.1. Algoritmi di Programmazione. Dividere e conquistare
 - 3.10.2. Algoritmi voraci
 - 3.10.3. Programmazione dinamica

Modulo 4. Sviluppo web multiplatforma per dispositivi mobili

- 4.1. Sviluppo Web Multiplatforma
 - 4.1.1. Sviluppo Web Multiplatforma
 - 4.1.2. App Ibride vs. App Native
 - 4.1.3. Tecnologie per la creazione di Apps ibride
- 4.2. *Progressive Web Apps (PWA)*
 - 4.2.1. *Progressive Web Apps (PWA)*
 - 4.2.2. *Progressive Web Apps (PWA)*. Caratteristiche
 - 4.2.3. *Progressive Web Apps (PWA)*. Costruzione
 - 4.2.4. *Progressive Web Apps (PWA)*. Limiti
- 4.3. *Framework Ionic*
 - 4.3.1. *Framework Ionic*. Analisi
 - 4.3.2. *Framework Ionic*. Caratteristiche
 - 4.3.3. Costruzione di una App con *Ionic*
- 4.4. *Framework* di sviluppo Web
 - 4.4.1. Analisi delle *Framework* nello Sviluppo Web
 - 4.4.2. *Framework* di Sviluppo Web
 - 4.4.3. Confronto tra *Frameworks* web
- 4.5. *Framework Angular*
 - 4.5.1. *Framework Angular*
 - 4.5.2. Uso di Angular nello Sviluppo di Applicazioni Multiplatforma
 - 4.5.3. Angular + *Ionic*
 - 4.5.4. Costruzione di Apps in Angular

- 4.6. Libreria di sviluppo *React*
 - 4.6.1. Libreria di *JavaScript React*
 - 4.6.2. Libreria di *JavaScript React*. Uso
 - 4.6.3. *React Native*
 - 4.6.4. *React + Ionic*
 - 4.6.5. Costruzione di App in *React*
- 4.7. *Framework* di Sviluppo *Vue*
 - 4.7.1. *Framework* di Sviluppo *Vue*
 - 4.7.2. *Framework* di Sviluppo *Vue* Uso
 - 4.7.3. *Vue + Ionic*
 - 4.7.4. Costruzione di Apps in *Vue*
- 4.8. *Framework* di Sviluppo *Electron*
 - 4.8.1. *Framework* di Sviluppo *Electron*
 - 4.8.2. *Framework* di Sviluppo *Electron*. Uso
 - 4.8.3. Distribuire le nostre Apps anche su desktop
- 4.9. Strumento di sviluppo per dispositivi mobili *Flutter*
 - 4.9.1. Strumento di sviluppo per dispositivi mobili *Flutter*
 - 4.9.2. Utilizzo del SDK *Flutter*
 - 4.9.3. Costruzione di Apps in *Flutter*
- 4.10. Strumento di Sviluppo per dispositivi mobili. Confronto
 - 4.10.1. Strumenti per lo Sviluppo di Applicazioni sui dispositivi mobili
 - 4.10.2. *Flutter* vs. *Ionic*
 - 4.10.3. Selezione dello *Stack* più adatto per la creazione di una App

Modulo 5. Database per lo sviluppo di applicazioni per dispositivi mobili

- 5.1. Database su dispositivi mobili
 - 5.1.1. Persistenza dei dati nello sviluppo di applicazioni mobili
 - 5.1.2. Funzionalità di database per Applicazioni mobili
 - 5.1.3. SQL. *Structured Query Language*
- 5.2. Scelta del database per le applicazioni mobili
 - 5.2.1. Analisi guidata da database di Applicazioni su dispositivi mobili
 - 5.2.2. Categorie di database
 - 5.2.3. Panorami di database
- 5.3. Sviluppo con SQLite
 - 5.3.1. Database SQLite
 - 5.3.2. Distribuzione del Modello
 - 5.3.3. Connessione a SQLite
- 5.4. Sviluppo con Oracle Berkeley DB
 - 5.4.1. Database Berkeley DB
 - 5.4.2. Distribuzione del modello
 - 5.4.3. Connessione a Berkeley DB
- 5.5. Sviluppo con Realm
 - 5.5.1. Capacità del Realm
 - 5.5.2. Creazione del database in Realm
 - 5.5.3. Connessione al Realm
- 5.6. Sviluppo con CouchDB Lite
 - 5.6.1. Database CouchDB Lite
 - 5.6.2. Creare database con CouchDB Lite
 - 5.6.3. Connessione con CouchDB Lite
- 5.7. Sviluppo con database centralizzati MySQL
 - 5.7.1. Database MySQL
 - 5.7.2. Distribuzione del modello relazionale con MySQL
 - 5.7.3. Connessione a MySQL
- 5.8. Sviluppi centralizzati. Oracle, MS SQL Server, MongoDB
 - 5.8.1. Sviluppo con Oracle
 - 5.8.2. Sviluppo con MS SQL Server
 - 5.8.3. Sviluppo con MongoDB
- 5.9. Dati di tipo Grafico
 - 5.9.1. Database orientata al Grafico
 - 5.9.2. Creazione di database con Neo4j
 - 5.9.3. Connettersi a Neo4j dall'applicazione mobile
- 5.10. Ambienti con capacità di stoccaggio
 - 5.10.1. Sviluppi con Firebase
 - 5.10.2. Sviluppi con i Core Data
 - 5.10.3. Sviluppo con Visual Builder Cloud Service

Modulo 6. Sviluppo di applicazioni per sistemi iOS

- 6.1. Ambiente di Sviluppo Xcode
 - 6.1.1. Creazione di un progetto
 - 6.1.2. Impostazione di un Emulatore per la compilazione
 - 6.1.3. Impostazione di un telefono fisico per la compilazione
- 6.2. Linguaggio di Programmazione Swift
 - 6.2.1. Swift I: linguaggio di programmazione
 - 6.2.2. Swift II: funzioni e loops
 - 6.2.3. Swift III: lambdas e structs
- 6.3. Librerie e Cocoa Pods
 - 6.3.1. Pods: installazione
 - 6.3.2. Configurazione di Cocoa Pods
 - 6.3.3. Struttura di Cocoa Pods
- 6.4. Librerie: Api, Database e R.swift
 - 6.4.1. Alamofire
 - 6.4.2. Database SQL con GRDB
 - 6.4.3. R.swift
- 6.5. Progettazione dello schermo
 - 6.5.1. Progettazione con Storyboard
 - 6.5.2. Responsive design
 - 6.5.3. Progettazione di viste tramite codice e SwiftUI
- 6.6. Assemblaggio di una vista
 - 6.6.1. UIViewController e il suo ciclo di vita
 - 6.6.2. Interazione tra schermi diversi
 - 6.6.3. Tipi di transizioni e modi di fare
- 6.7. Sensori e localizzazione
 - 6.7.1. Accesso ai sensori
 - 6.7.2. Accesso alla posizione in primo piano
 - 6.7.3. Accesso alla posizione di sfondo

- 6.8. Architettura
 - 6.8.1. MVP
 - 6.8.2. VIPER
 - 6.8.3. Architettura di Sviluppo IOS
- 6.9. Monetizzazione e *Analytics*
 - 6.9.1. *Firebase Analytics*
 - 6.9.2. *Firebase Crashlytics*
 - 6.9.3. Monetizzazione e annunci con Google ADMob
- 6.10. App Store e versioning
 - 6.10.1. Impostazione di un account App Store
 - 6.10.2. Versioni di prova (*Test Flight*)
 - 6.10.3. Lancio in produzione

Modulo 7. Distribuzioni a integrazione continua per dispositivi mobili

- 7.1. DevSecOps
 - 7.1.1. DevSecOps. Uso
 - 7.1.2. Analizzatori statici
 - 7.1.3. Test di sicurezza con analisi dinamica
- 7.2. Monitoraggio continuo
 - 7.2.1. Monitoraggio continuo
 - 7.2.2. Monitoraggio Continuo. Analisi e Vantaggi
 - 7.2.3. Monitoraggio Continuo. Piattaforme
- 7.3. Implementazione
 - 7.3.1. Implementazione della macchina locale
 - 7.3.2. Implementazione della macchina condivisa
 - 7.3.3. Implementazione basata su servizi cloud
 - 7.3.4. Gestione della configurazione

Modulo 8. User experience su dispositivi mobili

- 8.1. *User experience*
 - 8.1.1. *Client experience*
 - 8.1.2. *Client experience*. Requisiti
 - 8.1.3. Bidirezionale con il cliente
- 8.2. *Client experience*. Obiettivi e attrezzature
 - 8.2.1. *Client experience*. Obiettivi e attrezzature
 - 8.2.2. Processi iterativi
 - 8.2.3. Informazione necessaria
- 8.3. Micro-interazioni
 - 8.3.1. Relazioni *End-to-end*
 - 8.3.2. Interazioni
 - 8.3.3. Omnicanales
- 8.4. Comportamento dell'utente
 - 8.4.1. Progetto di fondazione
 - 8.4.2. Analitica web e di sessione
 - 8.4.3. Esperti di Analisi
- 8.5. Stato dell'Arte della Tecnologia
 - 8.5.1. *Machine Learning*
 - 8.5.2. Blockchain
 - 8.5.3. Internet of Things
- 8.6. Componenti tecnici
 - 8.6.1. Componenti tecnici
 - 8.6.2. Componenti avanzati: dispositivi
 - 8.6.3. Componenti avanzati: diversi profili
- 8.7. Usabilità
 - 8.7.1. Euristiche di Nielsen
 - 8.7.2. Test utente
 - 8.7.3. Usabilità. Errori
- 8.8. Tecniche di UX. *User experience*
 - 8.8.1. Regole
 - 8.8.2. *Prototyping*
 - 8.8.3. Strumenti *Low-Code*



- 8.9. Strategia Visuale
 - 8.9.1. Progettazione del *User Interface*
 - 8.9.2. Lavoro di *User Interface* sul web
 - 8.9.3. Lavoro di *User Interface* sulle Applicazioni
- 8.10. *Developer Frameworks*
 - 8.10.1. *Frameworks* di CX
 - 8.10.2. *Frameworks* di UX
 - 8.10.3. *Frameworks* di UI

Modulo 9. Sicurezza sui dispositivi mobili

- 9.1. Architettura di sicurezza di dispositivi mobili
 - 9.1.1. Sicurezza fisica dei dispositivi
 - 9.1.2. Sicurezza di sistema operativo
 - 9.1.3. Sicurezza delle applicazioni
 - 9.1.4. Sicurezza dei dati
 - 9.1.5. Sicurezza delle comunicazioni
 - 9.1.6. Sicurezza dei dispositivi aziendali
- 9.2. Protezione dell'Hardware mobile
 - 9.2.1. Dispositivi mobili
 - 9.2.2. Dispositivi Wearables
 - 9.2.3. Automobili
 - 9.2.4. Dispositivi IOT
 - 9.2.5. Dispositivi TV
- 9.3. Protezione del Sistema Operativo
 - 9.3.1. Dispositivi mobili Android
 - 9.3.2. Dispositivi mobili Apple IOS
 - 9.3.3. Altri dispositivi mobili esistenti: Blackberry, etc.
 - 9.3.4. Dispositivi Wearables
 - 9.3.5. Sistemi operativi per autoveicoli
 - 9.3.6. Dispositivi mobili nel *Internet of Things* (IoT)
 - 9.3.7. Dispositivi *SmatTV*
- 9.4. Protezione delle applicazioni mobili
 - 9.4.1. Dispositivi mobili Android
 - 9.4.2. Dispositivi mobili Apple IOS
 - 9.4.3. Altri dispositivi mobili. Blackberry
 - 9.4.4. Dispositivi *Wearables*
 - 9.4.5. Sistemi operativi per autoveicoli
 - 9.4.6. Dispositivi mobili nel *Internet of Things* (IoT)
 - 9.4.7. Dispositivi *SmatTV*
- 9.5. Protezione dei dati nelle applicazioni mobili
 - 9.5.1. Dispositivi mobili Android
 - 9.5.2. Dispositivi mobili Apple IOS
 - 9.5.3. Altri dispositivi mobili. Blackberry
 - 9.5.4. Dispositivi *Wearables*
 - 9.5.5. Sistemi operativi per autoveicoli
 - 9.5.6. Dispositivi mobili nell' *Internet of Things* (IoT)
 - 9.5.7. Dispositivi *SmatTV*
- 9.6. Sicurezza nei *Market Places* mobili
 - 9.6.1. *Google Play* di Google
 - 9.6.2. *Play Store* di Apple
 - 9.6.3. Altri *Market Places*
 - 9.6.4. *Rooting* dei dispositivi mobili
- 9.7. Soluzioni di sicurezza multiplatforma
 - 9.7.1. *Mobile Device Management* (MDM) unico
 - 9.7.2. Tipi di soluzioni presenti sul mercato
 - 9.7.3. Protezione dei dispositivi tramite MDM (*Master Data Management*)
- 9.8. Sviluppo sicuro di applicazioni mobili
 - 9.8.1. Utilizzo di standard per uno sviluppo sicuro
 - 9.8.2. Gestione integrata dei test di sicurezza
 - 9.8.3. Distribuzione sicuro di applicazioni

- 9.9. Gestione delle autorizzazioni sui dispositivi mobili
 - 9.9.1. Sistema di permessi
 - 9.9.2. Processi in esecuzione nel kernel
 - 9.9.3. Filo di esecuzione ed eventi
- 9.10. Raccomandazioni di sicurezza per dispositivi mobili
 - 9.10.1. Raccomandazioni di NSA sui dispositivi mobili
 - 9.10.2. Raccomandazioni di INCIBE sui dispositivi mobili
 - 9.10.3. ISO 27001:2013 Allegato

Modulo 10. Linguaggio di programmazione Android

- 10.1. Piattaforma Android
 - 10.1.1. Piattaforma Android
 - 10.1.2. Sistema Operativo Android
 - 10.1.3. *Open Handset Alliance* nello sviluppo Android
- 10.2. Architettura Android
 - 10.2.1. Elementi architetturali di un Sistema Android
 - 10.2.2. Comunicazione tra elementi
 - 10.2.3. Estensibilità dell'architettura Android
 - 10.2.4. Gestione delle risorse macchine: Batteria e Memoria
 - 10.2.5. Emulatori Android
- 10.3. Nucleo Linux di Android
 - 10.3.1. Composizione del nucleo
 - 10.3.2. Elementi strutturali del nucleo
 - 10.3.3. La Macchina Virtuale Dalvik
 - 10.3.4. La Macchina Virtuale Android RunTime (ART.)
- 10.4. Librerie Native di Android
 - 10.4.1. Librerie Native di Android
 - 10.4.2. Librerie di compatibilità (*Support Library*)
 - 10.4.3. Librerie native ed estensibilità
- 10.5. Il Sistema di File e Dati in Android
 - 10.5.1. Struttura di un'applicazione Android tipica
 - 10.5.2. Sistema di archiviazione YAFFS2 e ext4
 - 10.5.3. Uso di SQLite e Room per la Gestione dei Dati
- 10.6. Sicurezza in Android
 - 10.6.1. Sistema di Permessi
 - 10.6.2. Firme digitali nei *Android Application Package* (apk)
 - 10.6.3. Processi in esecuzione nel kernel
 - 10.6.4. Filo di esecuzione ed Eventi
- 10.7. Componenti strutturali di un'applicazione standard
 - 10.7.1. Vista (*View*)
 - 10.7.2. Attività (*Activity*)
 - 10.7.3. Frammento (*Fragment*)
 - 10.7.4. Servizio (*Service*)
 - 10.7.5. Intenzione (*Intent*)
 - 10.7.6. *Broadcasts Receiver* e *Content Provider*
 - 10.7.7. Gestione dei dati e preferenze d'utente
- 10.8. Versioni Android
 - 10.8.1. Versioni Android
 - 10.8.2. Rilascio di versioni Android
 - 10.8.3. Dispersione di distribuzioni Android
 - 10.8.4. Android versus Apple IOS e altri sistemi mobile
- 10.9. Android per veicoli
 - 10.9.1. Android e il mondo delle automobili
 - 10.9.2. Elementi strutturali in un sistema Android per automobili
 - 10.9.3. Comunicazione tra dispositivi
- 10.10. Android in Domotica *Wearable* e in *Internet of Things* (IoT)
 - 10.10.1. Il mondo connesso
 - 10.10.2. Elementi strutturali in un sistema Android per la Domotica
 - 10.10.3. Elementi di Android *Wearable*
 - 10.10.4. Android sul *Internet of Things* (IoT)

Modulo 11. Frameworks nello sviluppo di applicazioni Android

- 11.1. *Frameworks* nello Sviluppo di Applicazioni Android
 - 11.1.1. *Framework* nello Sviluppo di Applicazioni Android
 - 11.1.2. *Frameworks*. Tipologie
 - 11.1.3. Scelta del *Framework* per il progetto
- 11.2. Attuazione di *Frameworks* in Android
 - 11.2.1. *Frameworks Core* di Android per Java/kotlin
 - 11.2.2. *Jetpack compose*
 - 11.2.3. *Frameworks* in altri Linguaggi
- 11.3. Sistemi di Gestione di Librerie in Sviluppo
 - 11.3.1. *Gradle*
 - 11.3.2. Automatizzazione con *Gradle*
 - 11.3.3. Strumenti dello Sviluppo Maven
- 11.4. Codice pulito
 - 11.4.1. Codice ordinato
 - 11.4.2. Preparazione del Codice in Applicazioni Android
 - 11.4.3. *Bikeshedding* e prioritizzazione
- 11.5. Schema di Sviluppo in Android
 - 11.5.1. Categorie di schemi
 - 11.5.2. Differenze tra schemi
 - 11.5.3. *Factory*, *Observer* e *Singleton*
- 11.6. MVP Modello, vista e presentatore
 - 11.6.1. MVC Modello, vista e presentatore
 - 11.6.2. Modello, vista e presentatore
 - 11.6.3. Esempio pratico: Pokemon Battle
- 11.7. MVVM. Modello, vista e *View Model*
 - 11.7.1. MVC vs MVVM
 - 11.7.2. Modello, vista e *View Model*
 - 11.7.3. Esempio Pratico. Pokemon Battle II
- 11.8. *Frameworks* e librerie più utilizzate in Android
 - 11.8.1. Librerie di interazione con API
 - 11.8.2. Librerie di conversione dei dati
 - 11.8.3. *Firebase* e *Firebase Analytics*

- 11.9. *Framework* visual di Android
 - 11.9.1. Ciclo di vita di un'applicazione Android
 - 11.9.2. Visual design in XML
 - 11.9.3. Design di elementi e animazioni in XML
- 11.10. *Frameworks* di Android in altri linguaggi
 - 11.10.1. *React Native*
 - 11.10.2. *Flutter*
 - 11.10.3. *ionic*

Modulo 12. Interfacce e layouts nello sviluppo di applicazioni Android

- 12.1. Ciclo di vita dell'Interfaccia in Android
 - 12.1.1. Ciclo di vita in Android
 - 12.1.2. Relazione processo-attività
 - 12.1.3. Persistenza di stato dell'applicazione
 - 12.1.4. Architettura *Clean* applicata ad Android
- 12.2. Visual nello Sviluppo di Applicazioni Android (*Views*)
 - 12.2.1. Livello di Presentazione dell'Architettura *Clean*
 - 12.2.2. *Recicler View*
 - 12.2.3. *Adapter View*
- 12.3. Design nello Sviluppo di Applicazioni Android (*Layouts*)
 - 12.3.1. *Layouts* in Android
 - 12.3.2. *Constraint Layout*
 - 12.3.3. Creazione di *Layouts* usando *Android Studio Layout Editor*
- 12.4. Animazioni nello Sviluppo di Applicazioni Android (*Animations*)
 - 12.4.1. Icone e immagini
 - 12.4.2. Transizioni
 - 12.4.3. Differenza tra Animazione di Proprietà e Animazione di Viste
- 12.5. Attività e Intenzioni nello Sviluppo di Applicazioni Android (*Activity* e *Intentions*)
 - 12.5.1. Intenzione esplicite e implicite
 - 12.5.2. Barra d'azione
 - 12.5.3. Comunicazione tra attività

- 12.6. Risorse alternative e di Sistema (*Material Design, Cardboard, ecc.*)
 - 12.6.1. Materiale di *Design* per Android
 - 12.6.2. Multimedia in Android
 - 12.6.3. Realtà Virtuale con *Google Cardboard for Android NDK*
- 12.7. Stili e Argomenti nello Sviluppo di Applicazioni Android
 - 12.7.1. Stili in un progetto Android
 - 12.7.2. Argomenti per il progetto Android
 - 12.7.3. Riutilizzo di Stili e Argomenti
- 12.8. Grafici, Touch Screen e Sensori
 - 12.8.1. Lavoro con Grafici avanzati
 - 12.8.2. Gestione di dispositivi con Touch Screen e con Tastiera
 - 12.8.3. Uso di Sensori del Dispositivo Android
- 12.9. Design per Realtà Aumentata
 - 12.9.1. Interfacce complesse usando la Camera
 - 12.9.2. Sensori di posizione e GPS nella Realtà Aumentata
 - 12.9.3. Presentazione in schermi non standard
 - 12.9.4. Errori e problemi comuni
- 12.10. Configurazione avanzata di interfaccia con *AndroidManifest.xml*
 - 12.10.1. La potenza del file manifesto di Android
 - 12.10.2. Design programmatico versus design dichiarativo
 - 12.10.3. Componenti chiave dell'archivio

Modulo 13. Linguaggio di Programmazione in Applicazioni Android. Kotlin

- 13.1. Linguaggio di programmazione Kotlin
 - 13.1.1. Linguaggio di programmazione Kotlin
 - 13.1.2. Linguaggio di programmazione Kotlin. Caratteristiche
 - 13.1.3. Funzionamento di un programma in Kotlin
- 13.2. Programmare in Kotlin
 - 13.2.1. Struttura di un programma in Kotlin
 - 13.2.2. Parole riservate e sintassi
 - 13.2.3. Scrivere per console e leggere *inputs* dell'utente-Hola Mundo

- 13.3. Tipi e Variabili in Kotlin
 - 13.3.1. Tipi e Variabili in Kotlin
 - 13.3.2. Dichiarazione di Variabili: Var vs. Val
 - 13.3.3. Operatori
 - 13.3.4. Conversioni di tipo
 - 13.3.5. *Array*
- 13.4. Controllo di Flusso in Kotlin
 - 13.4.1. Controllo di Flusso
 - 13.4.2. Espressioni condizionali
 - 13.4.3. Loop
- 13.5. Funzioni in Kotlin
 - 13.5.1. Funzioni in Kotlin
 - 13.5.2. Struttura di una funzione
 - 13.5.3. Funzioni di sensibilizzazione (*Scope functions*)
- 13.6. Classi e Oggetti in Kotlin
 - 13.6.1. Classi e Oggetti in Kotlin
 - 13.6.2. Classi
 - 13.6.3. Oggetti
 - 13.6.4. Costruttori e inizializzazione di proprietà
 - 13.6.5. Classi annidate e classi interne
 - 13.6.6. Classi di dati
- 13.7. Ereditarietà in Kotlin
 - 13.7.1. Ereditarietà
 - 13.7.2. Superclassi e sottoclassi
 - 13.7.3. Sovrascrittura e proprietà e funzioni
 - 13.7.4. Eredità e altri tipi di relazione tra classi
 - 13.7.5. Classi sigillate
 - 13.7.6. Enumerazione
- 13.8. Classi astratte e Interfacce in Kotlin
 - 13.8.1. Classi astratte e Interfacce
 - 13.8.2. Classi astratte
 - 13.8.3. Interfacce
 - 13.8.4. Validazione e conversioni di Tipo-Operatori *is, when, as*

- 13.9. Collezioni in Kotlin
 - 13.9.1. Collezioni in Kotlin
 - 13.9.2. Liste
 - 13.9.3. Insieme
 - 13.9.4. Mappa
- 13.10. Gestione delle eccezioni e valori null in Kotlin
 - 13.10.1. Gestione delle Eccezioni e Valori Null
 - 13.10.2. Valori Null, tipi *nullable* e *non-nullable*
 - 13.10.3. Eccezioni

Modulo 14. Linguaggio di programmazione in applicazioni Android. Kotlin avanzato. Generalità, programmazione funzionale e parallelismo

- 14.1. Generalità in Kotlin
 - 14.1.1. Generalità in Kotlin
 - 14.1.2. Generalità in collezioni, funzioni, classi e interfacce
 - 14.1.3. Covarianza e Contravarianza: *Out* o *In*
- 14.2. Funzioni Lambda in Kotlin
 - 14.2.1. Funzioni Lambdas
 - 14.2.2. Struttura di una funzione lambda
 - 14.2.3. Uso di funzioni lambdas
- 14.3. Funzioni di ordine superiore in Kotlin
 - 14.3.1. Funzioni di ordine superiore
 - 14.3.2. Funzioni di ordine superiore standard di Kotlin
 - 14.3.3. Collegamento delle chiamate di funzione
- 14.4. Estensioni in Kotlin
 - 14.4.1. Estensioni in Kotlin
 - 14.4.2. Funzioni d'estensione
 - 14.4.3. Proprietà d'estensione
 - 14.4.4. Oggetti accompagnatori

- 14.5. Schema *Delegazione* in Kotlin
 - 14.5.1. Schema *Delegation*
 - 14.5.2. *Delegation* in Kotlin
 - 14.5.3. Proprietà delegate
- 14.6. Annotazioni e Riflessioni in Kotlin
 - 14.6.1. Annotazioni e Riflessioni
 - 14.6.2. Annotazioni in Kotlin
 - 14.6.3. Riflessione in Kotlin
- 14.7. *Testing* in Kotlin
 - 14.7.1. *Testing* in Kotlin
 - 14.7.2. *Frameworks* e Librerie di *Testing* in Kotlin
 - 14.7.3. Kotest
- 14.8. Programmazione asincrona in Kotlin
 - 14.8.1. Programmazione asincrona
 - 14.8.2. Tecniche di programmazione asincrona in Kotlin
 - 14.8.3. Tecniche di Programmazione comparativa
- 14.9. Corruzione in Kotlin
 - 14.9.1. Corruzione
 - 14.9.2. Canali
 - 14.9.3. Contesto e *Dispatchers*
 - 14.9.4. Stato condiviso e consenso
 - 14.9.5. Gestione delle eccezioni in corruzioni
- 14.10. Ecosistema Kotlin
 - 14.10.1. Ecosistema Kotlin
 - 14.10.2. Librerie per Kotlin
 - 14.10.3. Strumenti per Kotlin

Modulo 15. Gestione dei dati sui dispositivi Android

- 15.1. Gestione dei Dati. Tipologia
 - 15.1.1. Gestione delle Dati sui dispositivi mobili
 - 15.1.2. Alternative per la Gestione dei Dati sui dispositivi Android
 - 15.1.3. Generazione di dati per lavoro per Intelligenza Artificiale e Analitica d'Uso
 - 15.1.4. Strumenti di misurazione di rendimento per un'ottima gestione dei dati
- 15.2. Gestione delle preferenze utente
 - 15.2.1. Tipi di dati inclusi negli archivi di preferenze
 - 15.2.2. Gestione delle preferenze utente
 - 15.2.3. Esportare le preferenze. Gestione dei permessi
- 15.3. Sistemi di stoccaggio di file
 - 15.3.1. Classificazione di sistema di archiviazione in dispositivi mobili
 - 15.3.2. Sistema interno di archiviazione
 - 15.3.3. Sistema esterno di archiviazione
- 15.4. Archivi JSON come archiviazione in Android
 - 15.4.1. Informazione non strutturata in archiviazione JSON
 - 15.4.2. Librerie per la Gestione dei Dati con JSON
 - 15.4.3. Utilizzo di JSON in Android. Raccomandazioni e ottimizzazioni
- 15.5. Archivi XML come Archiviazione in Android
 - 15.5.1. Formato XML in Android
 - 15.5.2. XML attraverso librerie SAX
 - 15.5.3. XML attraverso librerie DOM
- 15.6. Database SQLite
 - 15.6.1. Database razionale per la Gestione dei Dati
 - 15.6.2. Uso del database
 - 15.6.3. Metodi SQLite per la gestione dei dati
- 15.7. Uso avanzato di Base di Dati SQLite
 - 15.7.1. Recupero di fallimenti utilizzando transazioni in SQLite
 - 15.7.2. Uso di cache per accelerare l'accesso ai dati
 - 15.7.3. Database su mobili
- 15.8. Libreria Room
 - 15.8.1. Architettura della libreria Room
 - 15.8.2. Libreria Room. Funzionalità
 - 15.8.3. Libreria Room. Vantaggi e svantaggi

- 15.9. *Content Provider* per condividere informazioni
 - 15.9.1. *Content Provider* per condividere informazioni
 - 15.9.2. *Content Provider* in Android. Tecnico d'Uso
 - 15.9.3. Sicurezza in *Content Provider*
- 15.10. Raccolta di Dati in Internet Cloud
 - 15.10.1. Android e Sistemi di stoccaggio nella Nube (*Cloud*)
 - 15.10.2. Servizi SOAP e REST per Android
 - 15.10.3. Problematica dei Sistemi distribuiti
 - 15.10.4. Internet come copia di sicurezza dei dati dell'applicazione

Modulo 16. Strumenti dei Dispositivi Android

- 16.1. Gestione: strumenti "TO DO"
 - 16.1.1. Strumenti di mercato
 - 16.1.2. Strumenti di mercato. Confronto tra le funzionalità
 - 16.1.3. Strumenti di gestione. Differenze
- 16.2. MDM: gestione dei dispositivi aziendali
 - 16.2.1. Controllo dei dispositivi aziendali
 - 16.2.2. Analisi dei principali strumenti di mercato
 - 16.2.3. Scelta dello strumento
- 16.3. CRM: strumenti di mercato
 - 16.3.1. Analisi degli strumenti di mercato con applicazione Android
 - 16.3.2. Strumenti di Mercato. Efficienza
 - 16.3.3. Strumenti di Mercato. Usi
- 16.4. Droni con Android
 - 16.4.1. App per dispositivi Android per il controllo dei droni
 - 16.4.2. Controlli autonomi
 - 16.4.3. Uso del drone con Android
- 16.5. Android, valore aggiunto nelle piattaforme bancarie
 - 16.5.1. Android nelle piattaforme Bancarie
 - 16.5.2. Rischi e frodi da parte dei criminali informatici
 - 16.5.3. Utilizzo dei dispositivi mobili
- 16.6. *Brokering* su dispositivi mobili
 - 16.6.1. Strumenti di mercato e loro utilizzo
 - 16.6.2. Confronto tra gli strumenti
 - 16.6.3. Scelta dello strumento per ogni utilizzo
- 16.7. Strumenti di intrattenimento e istruzione
 - 16.7.1. Usi
 - 16.7.2. Strumenti di mercato
 - 16.7.3. Confronto delle funzionalità tra gli Strumenti di Sviluppo Android
- 16.8. IoT Android
 - 16.8.1. *Framework* e piattaforme di mercato
 - 16.8.2. Rischi e considerazioni su Android IoT
 - 16.8.3. Usi dell'IoT su Android
- 16.9. Efficienza del processo
 - 16.9.1. Analisi degli strumenti di mercato per la creazione di App
 - 16.9.2. Confronto tra strumenti per la creazione di applicazioni Android
 - 16.9.3. *Caso d'uso*
- 16.10. Le applicazioni più scaricate oggi
 - 16.10.1. Strumenti più scaricati oggi
 - 16.10.2. Raggruppamento per famiglie
 - 16.10.3. Usi primari, secondari e confronto con iOS

Modulo 17. Design responsive su Android

- 17.1. *Responsive design*
 - 17.1.1. Design Responsive
 - 17.1.2. Utilizzabilità, accessibilità e UX
 - 17.1.3. Design responsive Vantaggi e svantaggi
- 17.2. *Mobile vs Tablet vs Web vs Smartwatches*
 - 17.2.1. Diversi formati, differenti dimensioni e diverse necessità
 - 17.2.2. Problemi nel design
 - 17.2.3. Adaptive vs responsive
- 17.3. *Guide di stile*
 - 17.3.1. Guide di Stile. Utilità
 - 17.3.2. Materiale di *design*
 - 17.3.3. Guida di Stile personale
- 17.4. *Layouting flessibile*
 - 17.4.1. *Layouting* flessibile
 - 17.4.2. *Layouting* Basico
 - 17.4.3. *Layouting* in Grid
 - 17.4.4. *Layouting* con *Relative Layout*
 - 17.4.5. *Layouting* con *Constraint Layout*
- 17.5. *Risorse flessibili*
 - 17.5.1. Risorse flessibili
 - 17.5.2. Immagini
 - 17.5.3. 9 patch
 - 17.5.4. Risorse globali
- 17.6. *Navigazione flessibile*
 - 17.6.1. Navigazione flessibile
 - 17.6.2. Navigazione con *Activities*
 - 17.6.3. Navigazione con *Fragments*
- 17.7. *Strumenti esterni*
 - 17.7.1. Generatori automatici
 - 17.7.2. Strumenti di prototipazione
 - 17.7.3. Strumenti di progettazione

- 17.8. *Debug e Tests*
 - 17.8.1. *Debug layouts*
 - 17.8.2. *Tests* automatici
 - 17.8.3. Sviluppo basato su componenti
 - 17.8.4. Buone prassi di *Testing* e Prove
- 17.9. *Alternative a Android nativo I. Web Pages*
 - 17.9.1. Design in *WebView*
 - 17.9.2. *ChromeCustomTabs*
 - 17.9.3. *Debug e Tests* in *web pages*
- 17.10. *Alternative ad Android nativo II. Applicazioni ibride*
 - 17.10.1. *React/React Native*
 - 17.10.2. *Flutter* vs.
 - 17.10.3. *Ionic*
 - 17.10.4. Apache Cordova

Modulo 18. Marketing per applicazioni Android

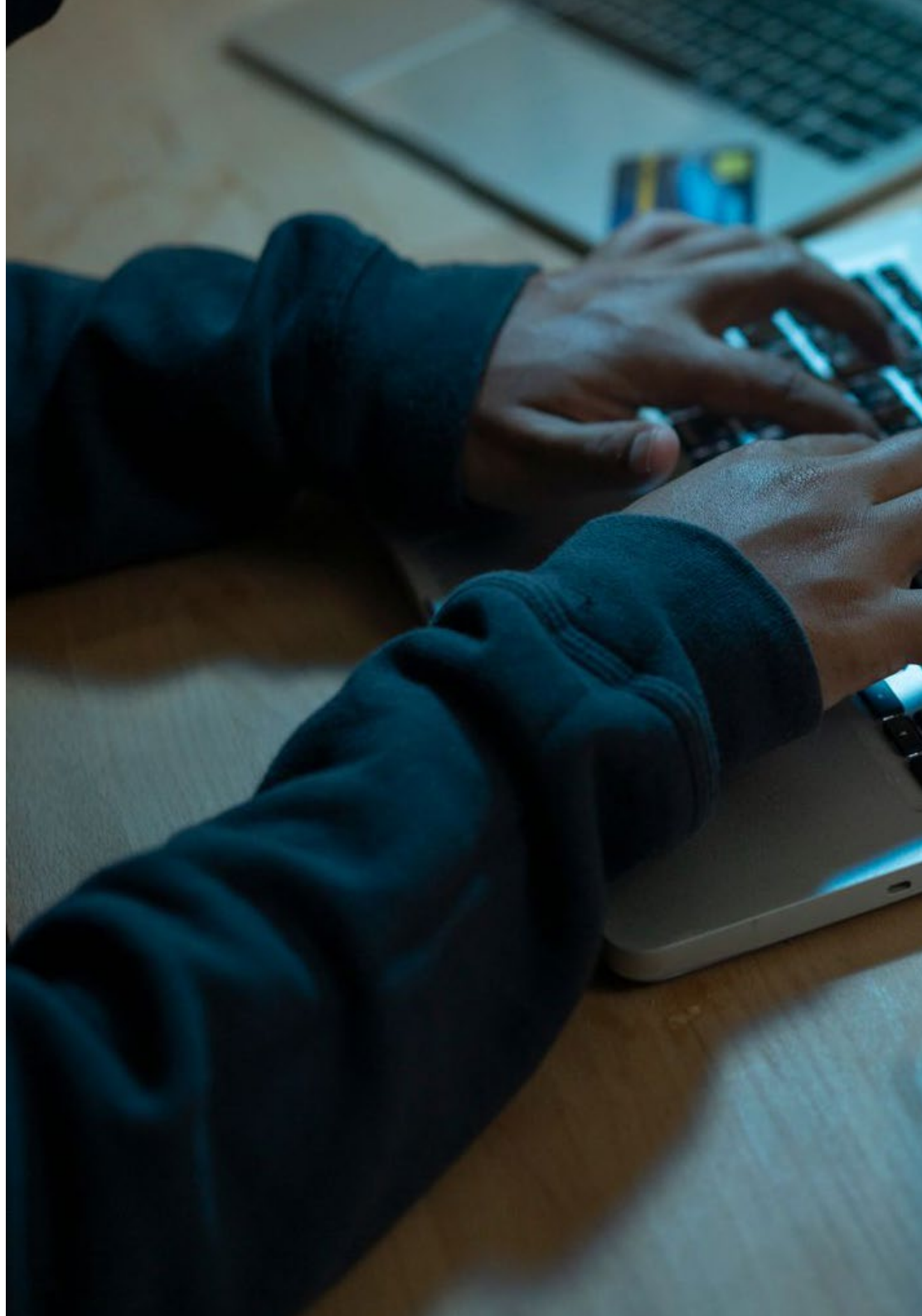
- 18.1. *Da Customer Service a Customer Experience*
 - 18.1.1. *Customer Service*. Sviluppo del cliente attuale
 - 18.1.2. Utente con accesso alle informazioni. Esigenze e necessità
 - 18.1.3. Il Feedback come fonte di conoscenze
- 18.2. *Customer Journey*
 - 18.2.1. Percorso dell'utente alla conversione
 - 18.2.2. Micro-segmentazione
 - 18.2.3. Esperienza attraverso i canali
- 18.3. *Misurazione della User Experience*
 - 18.3.1. Architettura web e mobile
 - 18.3.2. Analitica di sessione come nuovo standard
 - 18.3.3. Stato dell'Arte della esperienza dell'utente
- 18.4. *Marketing di Applicazioni Android*
 - 18.4.1. CX+IA
 - 18.4.2. CX+Blockchain
 - 18.4.3. CX+IoT

- 18.5. Prodotti di CX (esperienza del cliente)
 - 18.5.1. Standard del settore
 - 18.5.2. Presenza online
 - 18.5.3. Esperienza del cliente per tutti gli Agenti dello Sviluppo
- 18.6. Lavoro incentrato sull'utente
 - 18.6.1. Attrezzatura
 - 18.6.2. Pensiero di progettazione
 - 18.6.3. Lavoro sul campo
- 18.7. La scienza dell'utente
 - 18.7.1. La scienza dell'Utente. Regole d'oro
 - 18.7.2. Iterazione
 - 18.7.3. Errori più comuni
- 18.8. Prototipi e *Wireframing*
 - 18.8.1. Prototipi e *Wireframing*
 - 18.8.2. *Hands-on*
 - 18.8.3. Livello avanzato
- 18.9. Interfacce mobili
 - 18.9.1. Design visivo. Regole
 - 18.9.2. Interfacce di apps. Chiavi
 - 18.9.3. Buone prassi nello sviluppo di Interfacce Mobile
- 18.10. Buone pratiche per la Esperienza dell'Utente. Consigli per gli Sviluppatori
 - 18.10.1. Livello Uno. Buone prassi nel CX
 - 18.10.2. Livello Due. Buone prassi nel UX
 - 18.10.3. Livello Tre. Buone prassi nell'UI

Modulo 19. Ciclo di vita in applicazioni Android. Cloud, playstore e versioni

- 19.1. Ciclo di vita del *software*
 - 19.1.1. Ciclo di vita del *software*
 - 19.1.2. Metodologie Agili
 - 19.1.3. Il ciclo continuo del *software* agile
- 19.2. Sviluppo manuale del prodotto
 - 19.2.1. Integrazione manuale
 - 19.2.2. Consegna manuale
 - 19.2.3. Distribuzione manuale
- 19.3. Integrazione supervisionata
 - 19.3.1. Integrazione continua
 - 19.3.2. Revisione manuale
 - 19.3.3. Revisioni automatiche statiche
- 19.4. Prove logiche
 - 19.4.1. Test unitari
 - 19.4.2. Test di integrazione
 - 19.4.3. Test comportamentali
- 19.5. Integrazione continua
 - 19.5.1. Ciclo di integrazione continua
 - 19.5.2. Dipendenze tra integrazioni
 - 19.5.3. Integrazione continua come Metodologia di Gestione del Deposito

- 19.6. Consegna continua
 - 19.6.1. Consegna continua. Tipologie di problemi da risolvere
 - 19.6.2. Consegna continua. Risoluzione dei problemi
 - 19.6.3. Vantaggi degli Aggiornamenti Continui
- 19.7. Distribuzione continua
 - 19.7.1. Distribuzione Continua. Tipologie di Problemi da risolvere
 - 19.7.2. Distribuzione Continua. Risoluzione dei Problemi
- 19.8. *Firebase Test Lab*
 - 19.8.1. Configurazione da *GCloud*
 - 19.8.2. Configurazione di Jenkins
 - 19.8.3. Uso dei Jenkins. Vantaggi
- 19.9. Configurazione di *Gradle*
 - 19.9.1. Sistema di Automattizzazione *Gradle*
 - 19.9.2. Componente di *Gradle Build Flavors*
 - 19.9.3. Componente di *Gradle Linteo*
- 19.10. Ciclo di Vita in Applicazione Android. Esempio
 - 19.10.1. Configurazione di *SemaphoreCI* e *GitHub*
 - 19.10.2. Configurazione dei blocchi di lavoro
 - 19.10.3. Promozione e *Deployment*





“ *Ora inizia la strada verso un futuro migliore e tu ne fai parte. Inizia a progettare le migliori applicazioni mobili con questo Master Specialistico* ”

06

Metodologia

Questo programma ti offre un modo differente di imparare. La nostra metodologia si sviluppa in una modalità di apprendimento ciclico: ***il Relearning***.

Questo sistema di insegnamento viene applicato nelle più prestigiose facoltà di medicina del mondo ed è considerato uno dei più efficaci da importanti pubblicazioni come il ***New England Journal of Medicine***.



“

Scopri il Relearning, un sistema che abbandona l'apprendimento lineare convenzionale, per guidarti attraverso dei sistemi di insegnamento ciclici: una modalità di apprendimento che ha dimostrato la sua enorme efficacia, soprattutto nelle materie che richiedono la memorizzazione”

Caso di Studio per contestualizzare tutti i contenuti

Il nostro programma offre un metodo rivoluzionario per sviluppare le abilità e le conoscenze. Il nostro obiettivo è quello di rafforzare le competenze in un contesto mutevole, competitivo e altamente esigente.

“

Con TECH potrai sperimentare un modo di imparare che sta scuotendo le fondamenta delle università tradizionali in tutto il mondo”



Avrai accesso a un sistema di apprendimento basato sulla ripetizione, con un insegnamento naturale e progressivo durante tutto il programma.



Imparerai, attraverso attività collaborative e casi reali, la risoluzione di situazioni complesse in ambienti aziendali reali.

Un metodo di apprendimento innovativo e differente

Questo programma di TECH consiste in un insegnamento intensivo, creato ex novo, che propone le sfide e le decisioni più impegnative in questo campo, sia a livello nazionale che internazionale. Grazie a questa metodologia, la crescita personale e professionale viene potenziata, effettuando un passo decisivo verso il successo. Il metodo casistico, la tecnica che sta alla base di questi contenuti, garantisce il rispetto della realtà economica, sociale e professionale più attuali.

“

Il nostro programma ti prepara ad affrontare nuove sfide in ambienti incerti e a raggiungere il successo nella tua carriera”

Il Metodo Casistico è stato il sistema di apprendimento più usato nelle migliori Scuole di Informatica del mondo da quando esistono. Sviluppato nel 1912 affinché gli studenti di Diritto non imparassero la legge solo sulla base del contenuto teorico, il metodo casistico consisteva nel presentare loro situazioni reali e complesse per prendere decisioni informate e giudizi di valore su come risolverle. Nel 1924 fu stabilito come metodo di insegnamento standard ad Harvard.

Cosa dovrebbe fare un professionista per affrontare una determinata situazione?

Questa è la domanda con cui ti confrontiamo nel metodo dei casi, un metodo di apprendimento orientato all'azione. Durante il corso, gli studenti si confronteranno con diversi casi di vita reale. Dovranno integrare tutte le loro conoscenze, effettuare ricerche, argomentare e difendere le proprie idee e decisioni.

Metodologia Relearning

TECH coniuga efficacemente la metodologia del Caso di Studio con un sistema di apprendimento 100% online basato sulla ripetizione, che combina diversi elementi didattici in ogni lezione.

Potenziamo il Caso di Studio con il miglior metodo di insegnamento 100% online: il Relearning.

Nel 2019 abbiamo ottenuto i migliori risultati di apprendimento di tutte le università online del mondo.

In TECH imparerai con una metodologia all'avanguardia progettata per formare i manager del futuro. Questo metodo, all'avanguardia della pedagogia mondiale, si chiama Relearning.

La nostra università è l'unica autorizzata a utilizzare questo metodo di successo. Nel 2019, siamo riusciti a migliorare il livello di soddisfazione generale dei nostri studenti (qualità dell'insegnamento, qualità dei materiali, struttura del corso, obiettivi...) rispetto agli indicatori della migliore università online.



Nel nostro programma, l'apprendimento non è un processo lineare, ma avviene in una spirale (impariamo, disimpariamo, dimentichiamo e re-impariamo). Pertanto, combiniamo ciascuno di questi elementi in modo concentrico. Questa metodologia ha formato più di 650.000 laureati con un successo senza precedenti in campi diversi come la biochimica, la genetica, la chirurgia, il diritto internazionale, le competenze manageriali, le scienze sportive, la filosofia, il diritto, l'ingegneria, il giornalismo, la storia, i mercati e gli strumenti finanziari. Tutto questo in un ambiente molto esigente, con un corpo di studenti universitari con un alto profilo socio-economico e un'età media di 43,5 anni.

Il Relearning ti permetterà di apprendere con meno sforzo e più performance, impegnandoti maggiormente nella tua specializzazione, sviluppando uno spirito critico, difendendo gli argomenti e contrastando le opinioni: un'equazione diretta al successo.

Dalle ultime evidenze scientifiche nel campo delle neuroscienze, non solo sappiamo come organizzare le informazioni, le idee, le immagini e i ricordi, ma sappiamo che il luogo e il contesto in cui abbiamo imparato qualcosa è fondamentale per la nostra capacità di ricordarlo e immagazzinarlo nell'ippocampo, per conservarlo nella nostra memoria a lungo termine.

In questo modo, e in quello che si chiama Neurocognitive Context-dependent E-learning, i diversi elementi del nostro programma sono collegati al contesto in cui il partecipante sviluppa la sua pratica professionale.



Questo programma offre i migliori materiali didattici, preparati appositamente per i professionisti:



Materiali di studio

Tutti i contenuti didattici sono creati appositamente per il corso dagli specialisti che lo impartiranno, per fare in modo che lo sviluppo didattico sia davvero specifico e concreto.

Questi contenuti sono poi applicati al formato audiovisivo che supporterà la modalità di lavoro online di TECH. Tutto questo, con le ultime tecniche che offrono componenti di alta qualità in ognuno dei materiali che vengono messi a disposizione dello studente.



Master class

Esistono evidenze scientifiche sull'utilità dell'osservazione di esperti terzi.

Imparare da un esperto rafforza la conoscenza e la memoria, costruisce la fiducia nelle nostre future decisioni difficili.



Pratiche di competenze e competenze

Svolgerai attività per sviluppare competenze e capacità specifiche in ogni area tematica. Pratiche e dinamiche per acquisire e sviluppare le competenze e le abilità che uno specialista deve sviluppare nel quadro della globalizzazione in cui viviamo.



Letture complementari

Articoli recenti, documenti di consenso e linee guida internazionali, tra gli altri. Nella biblioteca virtuale di TECH potrai accedere a tutto il materiale necessario per completare la tua specializzazione.





Casi di Studio

Completerai una selezione dei migliori casi di studio scelti appositamente per questo corso. Casi presentati, analizzati e monitorati dai migliori specialisti del panorama internazionale.



Riepiloghi interattivi

Il team di TECH presenta i contenuti in modo accattivante e dinamico in pillole multimediali che includono audio, video, immagini, diagrammi e mappe concettuali per consolidare la conoscenza.

Questo esclusivo sistema di specializzazione per la presentazione di contenuti multimediali è stato premiato da Microsoft come "Caso di successo in Europa".



Testing & Retesting

Valutiamo e rivalutiamo periodicamente le tue conoscenze durante tutto il programma con attività ed esercizi di valutazione e autovalutazione, affinché tu possa verificare come raggiungi progressivamente i tuoi obiettivi.



07

Titolo

Il Master Specialistico in Sviluppo di Applicazioni Mobili, Esperto in Sistemi Android garantisce, oltre alla preparazione più rigorosa e aggiornata, il conseguimento di una qualifica di Master Specialistico rilasciata da TECH Università Tecnologica.



“

Porta a termine questo programma e ricevi la tua qualifica universitaria senza spostamenti o fastidiose formalità”

Questo **Master Specialistico in Sviluppo di Applicazioni Mobili, Esperto in Sistemi Android** possiede il programma più completo e aggiornato del mercato.

Dopo aver superato la valutazione, lo studente riceverà mediante lettera certificata* con ricevuta di ritorno, la sua corrispondente qualifica di **Master Specialistico** rilasciata da **TECH Università Tecnologica**.

Il titolo rilasciato da **TECH Università Tecnologica** esprime la qualifica ottenuta nel Master Specialistico, e riunisce tutti i requisiti comunemente richiesti da borse di lavoro, concorsi e commissioni di valutazione di carriere professionali.

Titolo: **Master Specialistico in Sviluppo di Applicazioni Mobili, Esperto in Sistemi Android**

N° Ore Ufficiali: **3000 o.**



*Apostille dell'Aia. Se lo studente dovesse richiedere che il suo diploma cartaceo sia provvisto di Apostille dell'Aia, TECH EDUCATION effettuerà le gestioni opportune per ottenerla pagando un costo aggiuntivo.

futuro
salute fiducia persone
educazione informazione tutor
garanzia accreditamento insegnamento
istituzioni tecnologia apprendimento
comunità impegno
attenzione personalizzata innovazione
conoscenza presente qualità
formazione online
sviluppo istituzioni
classe virtuale lingue

tech università
tecnologica

Master Specialistico
Sviluppo di Applicazioni
Mobili, Esperto in
Sistemi Android

- » Modalità: online
- » Durata: 2 anni
- » Titolo: TECH Università
Tecnologica
- » Orario: a scelta
- » Esami: online

Master Specialistico

Sviluppo di Applicazioni Mobili,
Esperto in Sistemi Android