

Mastère Avancé

Développement d'Applications
Mobiles, Expert Android



Mastère Avancé

Développement d'Applications Mobiles, Expert Android

- » Modalité: en ligne
- » Durée: 2 ans
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

Accès au site web: www.techtitute.com/fr/mastere-avance/mastere-avance-developpement-applications-mobiles-expert-android

Accueil

01

Présentation

page 4

02

Objectifs

page 8

03

Compétences

page 18

04

Direction de la formation

page 22

05

Structure et contenu

page 30

06

Méthodologie

page 50

07

Diplôme

page 58

01 Présentation

Le monde d'aujourd'hui est inconcevable sans les applications et les entreprises qui tirent parti des nouvelles technologies. Les applications mobiles constituent un progrès pour toute entreprise, un signe de maturité et un effort d'adaptation aux nouvelles habitudes des utilisateurs, ce qui les rend indispensables dans pratiquement tous les domaines. Aujourd'hui, le besoin d'applications mobiles est lié aux fonctionnalités qu'elles peuvent offrir au-delà des informations qu'une page web peut posséder. En conséquence, il y a aujourd'hui plus de 3.000 millions de dispositifs Android dans le monde et, par conséquent, le secteur du Développement d'Applications Mobiles a connu une croissance rapide. C'est maintenant, plus que jamais, que l'industrie a un besoin urgent de professionnels spécialisés dans le Développement d'Applications, ce qui a motivé la création de ce programme.





“

Apprenez auprès des experts tous les aspects clés de l'expérience client pour développer des applications sur Android avec une forte entrée sur le marché"

Les rapports sur le marché du travail confirment la demande croissante de profils d'experts en conception d'applications mobiles, valorisant l'expertise dans le cycle de vie complet du développement, du déploiement et de la monétisation des applications. L'objectif est de fournir à l'informaticien toutes les connaissances sur le langage de programmation d'applications, avec un accent particulier sur Android, les architectures et les interfaces utilisateur. Ce programme a été conçu par les meilleurs experts en la matière, qui constituent le corps enseignant.

Afin, d'offrir à l'étudiant les compétences nécessaires pour comprendre les différents langages de programmation pour chaque type d'appareil, en approfondissant la conception *responsive*. Cela leur permettra de développer des applications adaptables aux différentes caractéristiques de chaque appareil. À son tour, le professionnel approfondira des domaines tels que l'informatique web, les logiciels, le développement commercial, le Marketing et les ventes, le conseil, la technologie mobile multiplateforme, les entreprises, le monde de l'automobile, la domotique, l'Internet des Objets (IoT), la banque et les drones.

En outre, TECH dispose de la meilleure méthodologie d'étude 100% en ligne, ce qui élimine la nécessité d'assister cours en personne ou de devoir se conformer à un horaire prédéterminé. Ainsi, en 24 mois, les informaticiens acquerront les bases nécessaires pour créer leur propre entreprise basée sur le développement d'applications ou pour entreprendre des tâches de conseil dans tous les aspects liés à la technologie mobile sur Android.

Le **Mastère Avancé en Développement d'Applications Mobiles, Expert Android** contient le programme académique le plus complet et le plus actuel du marché. Les principales caractéristiques sont les suivantes:

- ◆ Le développement d'études de cas présentées par des experts en Informatique
- ◆ Le contenu graphique, schématique et éminemment pratique du programme fournit des informations scientifiques et pratiques sur les disciplines essentielles à la pratique professionnelle
- ◆ Les exercices pratiques où effectuer le processus d'auto-évaluation pour améliorer l'apprentissage
- ◆ L'accent est mis sur les méthodologies innovantes pour le développement d'applications mobiles
- ◆ Cours théoriques, questions à l'expert, forums de discussion sur des sujets controversés et travail de réflexion individuel
- ◆ La disponibilité de l'accès au contenu à partir de n'importe quel appareil fixe ou portable doté d'une connexion à internet



Dans ce Mastère Avancé, vous serez à jour et vous connaîtrez en profondeur les différents langages de programmation pour chaque type d'appareil mobile"

“

Au fur et à mesure que vous maîtriser tout le contenu sur le Développement d'Applications Mobiles, vous comprendrez également comment les monétiser et approfondir le Marketing Mobile"

Son corps enseignant comprend des professionnels du domaine de l'Informatique, qui apportent leur expérience professionnelle à ce programme, ainsi que des spécialistes reconnus de grandes sociétés et d'universités prestigieuses.

Son contenu multimédia, développé avec les dernières technologies éducatives, permettra au professionnel un apprentissage situé et contextuel, c'est-à-dire un environnement simulé qui fournira un étude immersif programmé pour s'entraîner dans des situations réelles.

La conception de ce programme est axée sur l'Apprentissage par les Problèmes, grâce auquel l'étudiant doit essayer de résoudre les différentes situations de la pratique professionnelle qui se présentent tout au long du programme. Pour ce faire, le professionnel aura l'aide d'un système vidéo interactif innovant créé par des experts reconnus.

Offrez des solutions efficaces à votre entreprise, avec la gestion des fondamentaux liés au Développement d'Applications dans les Systèmes Android.

TECH vous donne l'occasion d'apprendre à développer des applications mobiles de manière autonome et professionnelle, sur des appareils multiplateformes.



02 Objectifs

En raison de la croissance et de la consolidation notoires qu'entraîne le monde des technologies et des appareils mobiles, l'informaticien est obligé d'actualiser continuellement ses connaissances afin d'acquérir tous les savoirs nécessaires. Par conséquent, ce programme vous fournit tous les outils et fondamentaux nécessaires pour approfondir le développement d'applications mobiles, avec un accent particulier sur le système Android, et pour fournir des solutions efficaces dans l'environnement professionnel. Les titulaires de ce diplôme seront au fait des tendances et sauront comment agir dans tous les domaines, allant de la conception et de la programmation à l'expérience de l'utilisateur final.



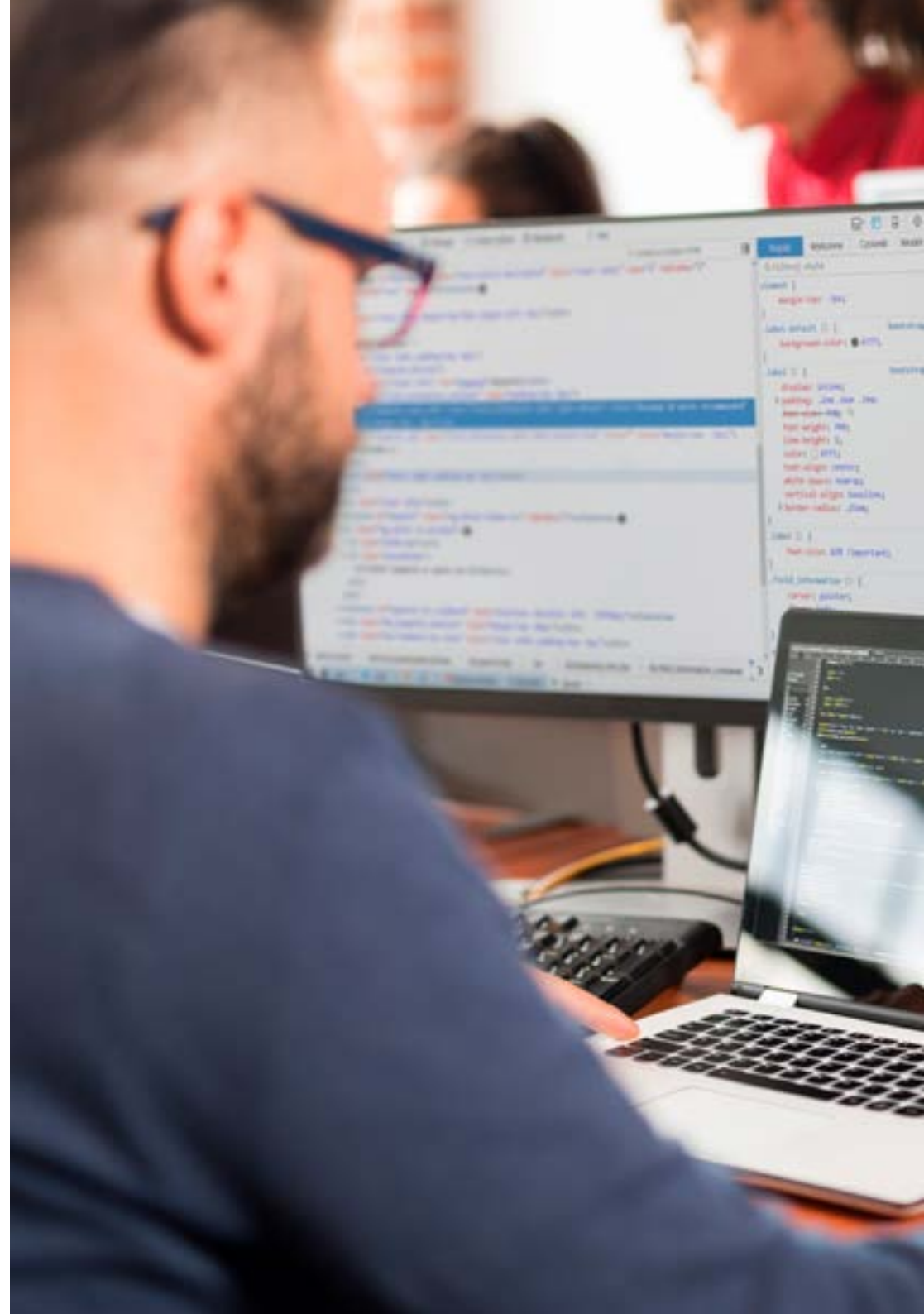
“

Ne manquez pas l'occasion de mettre à jour vos connaissances avec ce Mastère Avancé et d'être en mesure de fournir des solutions efficaces dans l'environnement professionnel"



Objectifs généraux

- ◆ Analyser les besoins et le comportement des utilisateurs par rapport aux appareils mobiles et à leurs applications
- ◆ Exécuter la conception d'architectures, d'itérations et d'interfaces utilisateurs à travers les langages de programmation des plateformes mobiles les plus représentatives du marché (Web, iOS et Android)
- ◆ Appliquer des mécanismes de contrôle d'erreur, de test et de débogage dans le développement d'applications mobiles
- ◆ Aborder les différents cas pratiques et commerciaux de publication, de distribution et de diffusion d'applications mobiles sur les principaux marchés d'applications
- ◆ Maîtriser les connaissances pratiques pour planifier et gérer des projets technologiques liés aux technologies mobiles
- ◆ Développer les compétences, aptitudes et outils nécessaires pour apprendre à développer des applications mobiles de manière autonome et professionnelle, sur des appareils multiplateformes
- ◆ Explorez les contenus liés à la monétisation des applications et au Marketing Mobile
- ◆ Déterminer les éléments structurels d'un Système Android
- ◆ Analyser les différences entre les différents *Frameworks* de développement, leurs forces et leurs faiblesses
- ◆ Développer des capacités avancées et des bonnes pratiques en matière de développement d'applications dans le langage de programmation Kotlin
- ◆ Développer une méthodologie pour une gestion optimale des données dans l'appareil
- ◆ Analyser les cas d'utilisation des appareils Android sur le marché
- ◆ Maîtriser les éléments du responsive design et relever les défis du Responsive design
- ◆ Compiler les différentes étapes d'un cycle d'intégration continue et axé sur le développement Android





Objectifs spécifiques

Module 1. Méthodologies de programmation dans le développement d'applications pour mobiles

- ◆ Explorer les processus traditionnels de développement de logiciels
- ◆ Analyser les processus de développement agiles
- ◆ Promouvoir les pratiques de développement
- ◆ Examiner les différentes techniques de représentation et de diagramme
- ◆ Découvrez les différents modèles de conception présents dans l'industrie du logiciel
- ◆ Explorer les différentes techniques de test de logiciels
- ◆ Reconnaître les normes et standards de référence de qualité dans le développement

Module 2. Technologies de développement d'applications pour mobiles

- ◆ Établir des concepts de dispositifs mobiles
- ◆ Compiler les plateformes principales
- ◆ Examiner leurs composants communs
- ◆ Identifier les composants différenciés, leurs capacités et leurs limites
- ◆ Définir les différents scénarios dans lesquels ils peuvent opérer Avantages
- ◆ Analyser les différentes interactions que ces dispositifs peuvent médiatiser
- ◆ Sensibiliser aux différents abus qui peuvent être commis

Module 3. Outils de travail pour le développement d'applications mobiles

- ◆ Préparer l'environnement de développement
- ◆ Acquérir des compétences en matière de terminal de commande
- ◆ Utiliser efficacement le système de contrôle de version
- ◆ Traiter de l'utilisation des systèmes de version de code à distance
- ◆ Établir les notions clés du fonctionnement d'internet
- ◆ Développer des concepts pertinents de programmation de logiciels
- ◆ Examiner les structures de données
- ◆ Examiner les techniques de conception et d'interprétation des algorithmes

Module 4. Développement web multiplateforme pour mobiles

- ◆ Déterminer les avantages et les limites du modèle de développement natif et du développement d'applications hybrides
- ◆ Examen des caractéristiques et des limites des applications Web progressives (PWA)
- ◆ Analyser les principaux *Frameworks* pour le développement d'applications web: Angular, React, Vue
- ◆ Compiler les principales technologies pour le développement d'applications mobiles multiplateformes: *Ionic* et *Flutter*
- ◆ Analysez les possibilités de déployer ces applications hybrides en tant qu'applications Web ou de bureau sur les PC
- ◆ Examiner un modèle afin de choisir l'alternative la plus appropriée pour le développement d'une application spécifique

Module 5. Base de données pour le développement d'applications mobiles

- ◆ Identifier le meilleur modèle de base de données en fonction des caractéristiques de l'application mobile
- ◆ Établir les capacités de chaque système de base de données
- ◆ Déterminer les différences entre les différentes bases de données
- ◆ Examiner comment se connecter et charger/extraire des données de différents types de bases de données
- ◆ Analyser les capacités de base des environnements de développement avec des capacités de base de données intégrées

Module 6. Développement d'applications pour les systèmes iOS

- ◆ Développer une application en Swift
- ◆ Utilisation des pods Cocoa pour gérer les bibliothèques
- ◆ Utilisation d'Alamofire pour connecter notre application à une API RESTful
- ◆ Préciser les exigences de base pour le choix d'une bibliothèque
- ◆ Développer un système de monétisation avec ADMOB
- ◆ Concevoir des vues à partir du code
- ◆ Publier une application dans l'App Store

Module 7. Déploiements d'intégration continue pour mobiles

- ◆ Déterminez le scénario le plus défavorable qui rend nécessaire l'application de cette méthode
- ◆ Spécifier les exigences auxquelles le logiciel à intégrer doit répondre
- ◆ Définir ce que sont l'intégration continue, la livraison continue et le déploiement continu
- ◆ Analyser DevSecOps
- ◆ Examiner la surveillance continue
- ◆ Développer les mises en œuvre des différentes étapes

Module 8. User Experience dans les dispositifs Mobiles

- ◆ Analyser le nouveau type d'utilisateur, ses interactions et son parcours dans les applications mobiles et les sites web
- ◆ Déterminer les outils fondamentaux de l'analyse web, de la mobilité et de l'accessibilité
- ◆ Préciser les techniques d'évaluation des micro-interactions et la conception d'expériences personnalisées
- ◆ Établir comment les nouvelles technologies perturbatrices telles que l'IA ou l'IdO ont fait passer l'expérience client à de nouveaux standards
- ◆ Montrer comment l'analyse comportementale génère une quantité et une qualité de données inédites dans l'analyse traditionnelle
- ◆ Développer de nouvelles méthodologies, telles que le *Design Thinking*, centrées sur l'utilisateur
- ◆ Proposer des outils basiques et avancés de prototypage et de *Wireframing*

Module 9. Sécurité des appareils mobiles

- ◆ Déterminer les éléments et les niveaux de sécurité d'un appareil mobile
- ◆ Établir les techniques à utiliser pour un dispositif
- ◆ Analyser les pièges courants en matière de sécurité
- ◆ Examiner les mécanismes activés dans la programmation pour prévenir les failles de sécurité
- ◆ Préciser les recommandations données par les organismes de sécurité
- ◆ Compiler les différentes solutions disponibles sur le marché pour gérer la sécurité des appareils mobiles au sein des entreprises
- ◆ Analyser les processus cryptographiques appliqués à la sécurité mobile

Module 10. Langage de programmation Android

- ◆ Examen du noyau Linux et de la machine virtuelle sur la base Android
- ◆ Analyser les bibliothèques du système natif
- ◆ Établir les points forts d'Android par rapport aux autres plateformes
- ◆ Déterminer les éléments d'une application Android
- ◆ Présentation des versions d'Android et de leurs améliorations
- ◆ Évaluation du marché des applications Android
- ◆ Fonder l'évolution future d'Android

Module 11. Frameworks utilisés dans le développement d'applications Android

- ◆ Analyser le *Framework* de Android Core
- ◆ Développer d'autres *Framework* utilisés pour le développement d'applications Android
- ◆ Implémentation de bibliothèques avec gradle
- ◆ Mettre en place des *Frameworks* pour se connecter à une API
- ◆ Générer des connaissances spécialisées sur Architectures telles que MVP
- ◆ Clarifier les avantages et les inconvénients de MVP et MVVM

Module 12. Interfaces et *layouts* dans le développement d'applications Android

- ◆ Présentation du modèle de cycle de vie des vues dans Android
- ◆ Examiner les attributs les plus importants d'une conception visuelle (*layout*)
- ◆ Analyser les modèles (*layout*) disponibles
- ◆ Générer une mise en page (*layout*) réutilisable
- ◆ Déterminer comment utiliser les ressources alternatives
- ◆ Identifier les différences d'utilisation de ces composants par rapport à d'autres systèmes de programmation
- ◆ Établir le potentiel et l'utilisation du fichier *AndroidManifest.xml*

Module 13. Langage de programmation les applications Android. Kotlin

- ◆ Développer le langage langage de programmation Kotlin
- ◆ Compiler les caractéristiques et les capacités différentielles du langage Kotlin
- ◆ Examiner le modèle d'exécution de base d'un programme Kotlin
- ◆ Analyser la syntaxe du langage et la structure d'un programme Kotlin
- ◆ Préciser le modèle des types et des variables en Kotlin
- ◆ Établir les différentes formes de gestion du flux de code en Kotlin
- ◆ Déterminer le modèle des classes, des collections et des objets en Kotlin
- ◆ Générer des connaissances expertes sur le modèle d'héritage en Kotlin
- ◆ Spécifier le modèle de gestion des exceptions et des types nuls en Kotlin

Module 14. Langage de programmation les applications Android. Kotlin avancé Généricité, programmation fonctionnelle et parallélisme

- ◆ Examiner le modèle de généricité covariante et contravariante en Kotlin
- ◆ Analyser le modèle de programmation fonctionnelle avec *Lambdas* de Kotlin
- ◆ Concrétiser les fonctions d'ordre supérieur de Kotlin
- ◆ Développer des extensions et des objets compagnons en Kotlin
- ◆ Examiner l'utilisation du modèle de *Delegation* en Kotlin
- ◆ Compiler les annotations et la réflexion en Kotlin
- ◆ Approfondir le modèle de *Testing* en Kotlin
- ◆ Établir les différents modèles de programmation asynchrone disponibles en Kotlin
- ◆ Déterminer le modèle des coroutines en Kotlin
- ◆ Compiler les différentes bibliothèques et outils utilitaires de l'écosystème Kotlin

Module 15. Gestion des données sur les appareils Android

- ♦ Analyser les différentes techniques de gestion des données sur les appareils Android
- ♦ Proposer des méthodes pour une utilisation optimale des données sur l'appareil
- ♦ Identifier les outils nécessaires à l'optimisation des données
- ♦ Examiner les caractéristiques de JSON et XML pour la gestion des données dans Android
- ♦ Évaluer les questions générales relatives aux systèmes distribués applicables au monde des applications pour appareils mobiles
- ♦ Déterminer l'utilisation de la bibliothèque *Room* en tant qu'abstraction pour l'utilisation de SQLite sur Android, ainsi que ses avantages et ses inconvénients
- ♦ Établir les permissions de sécurité nécessaires à la gestion des données dans n'importe laquelle des techniques utilisées dans Android

Module 16. Outils pour les appareils Android

- ♦ Compiler les outils les plus avancés dans la gestion quotidienne
- ♦ Évaluer les outils de contrôle des appareils Android
- ♦ Démontrer les avantages de l'utilisation d'Android sur les drones
- ♦ Préciser les fonctionnalités des outils de gestion CRM
- ♦ Démontrer les avantages des appareils Android dans les plateformes Bancaires
- ♦ Analyser les possibilités de l'IdO et des plateformes Android
- ♦ Examiner les outils d'efficacité des processus sur Android

Module 17. Conception *responsive* dans Android

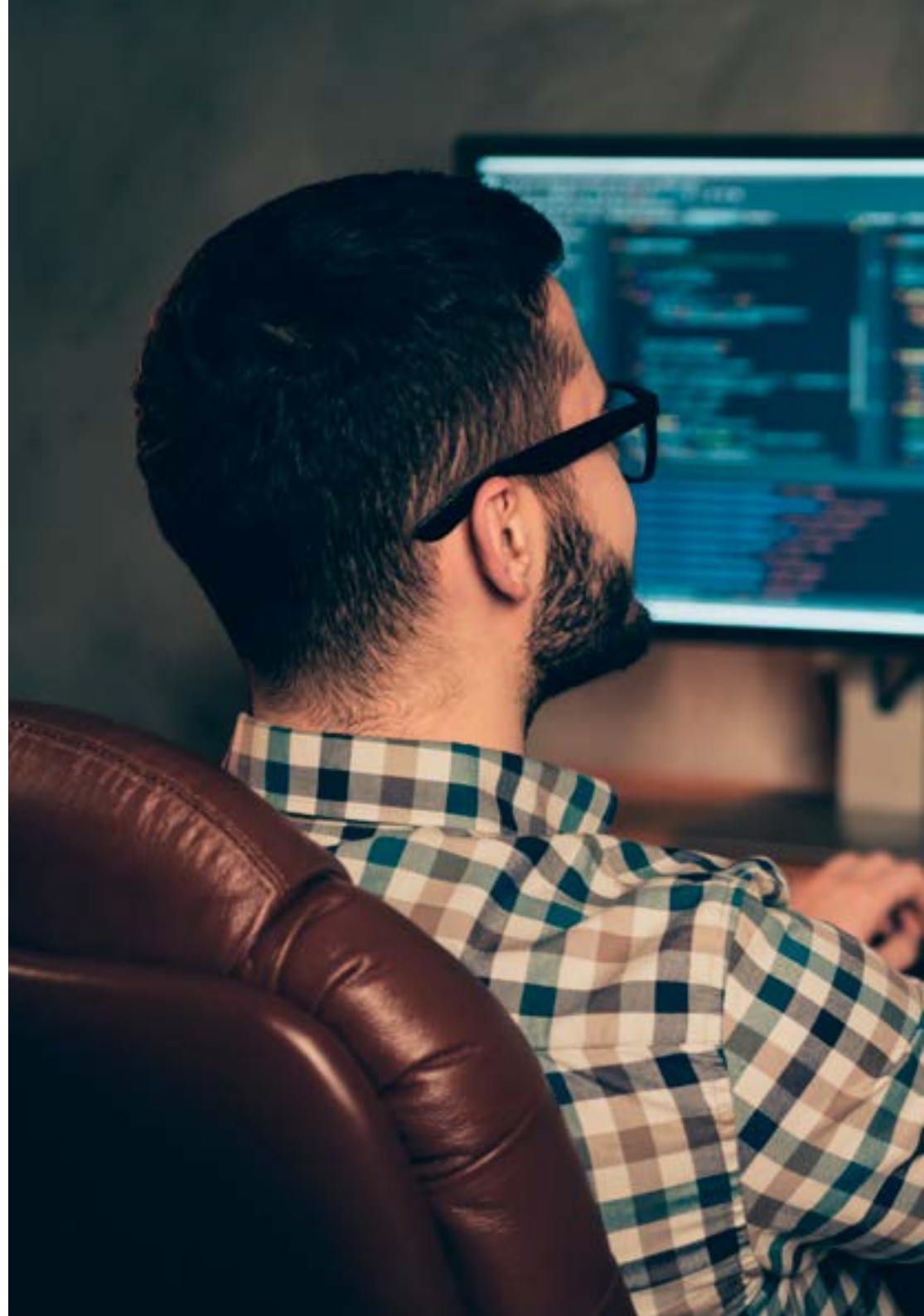
- ♦ Analyser les principaux éléments d'un design
- ♦ Définir une méthodologie de conception pour les éléments visuels et une méthodologie de conception pour les écrans
- ♦ Résoudre les différents problèmes rencontrés dans le développement multi-appareils
- ♦ Disposer d'outils pour générer de meilleures et plus nombreuses ressources pour le développement multi-appareils
- ♦ Examiner *Frameworks* alternatifs au développement réactif natif
- ♦ Générer une méthodologie propriétaire pour développer des applications en utilisant les meilleures pratiques pour l'affichage multi-appareils dès le début du projet

Module 18. Marketing dans les applications Android

- ♦ Analyser les Nouvelles méthodologies et outils centrés sur l'utilisateur
- ♦ Déterminer comment l'Intelligence Artificielle a porté le CX au niveau supérieur
- ♦ Établir l'importance de l'accessibilité et de la mobilité
- ♦ Développer des techniques de pointe pour l'analyse des sessions et des comportements
- ♦ Spécifier les objectifs de micro-personnalisation au cours du *journey* de l'utilisateur
- ♦ Compiler de nouvelles méthodologies pour un environnement changeant et vivant
- ♦ Proposer des techniques de prototypage

Module 19. Cycle de vie d'une application Android. Cloud, playstore et versioning

- ◆ Concrétiser les avantages de l'adoption d'un modèle automatisé de déploiement des versions
- ◆ Différencier entre intégration continue, livraison et déploiement continus
- ◆ Définir des principales fonctionnalités de DevOps
- ◆ Évaluer certains des outils essentiels pour implémenter les pipelines CI/CD
- ◆ Concrétiser les facteurs essentiels pour développer des applications prêtes à supporter les processus CI/CD
- ◆ Examiner les technologies de conteneurs comme un pilier fondamental de la pratique du CI/CD
- ◆ Identifier les pratiques, les cas d'utilisation, les technologies et les outils de l'écosystème CI/CD





“

Vous acquerez des connaissances spécialisées sur le cycle de vie des applications Android afin de travailler plus efficacement en tant que développeur”

03

Compétences

La demande en industrie technologique de profils professionnels hors pair dans le secteur des technologies implique une formation constante pour être performant sur un marché à l'évolution vertigineuse. Ce Mastère Avancé forme des informaticiens ayant une connaissance spécialisée et actualisée des différents composants technologiques des appareils mobiles pour développer des applications. Le diplômé aura une connaissance approfondie des meilleurs *Hardwares* et *Softwares* indispensables à tout professionnel travaillant dans ce secteur. De cette manière, l'informaticien sera à jour dans les connaissances nécessaires pour être autonome lors du développement d'une application pour les appareils Android, allant de la conception à la production.



“

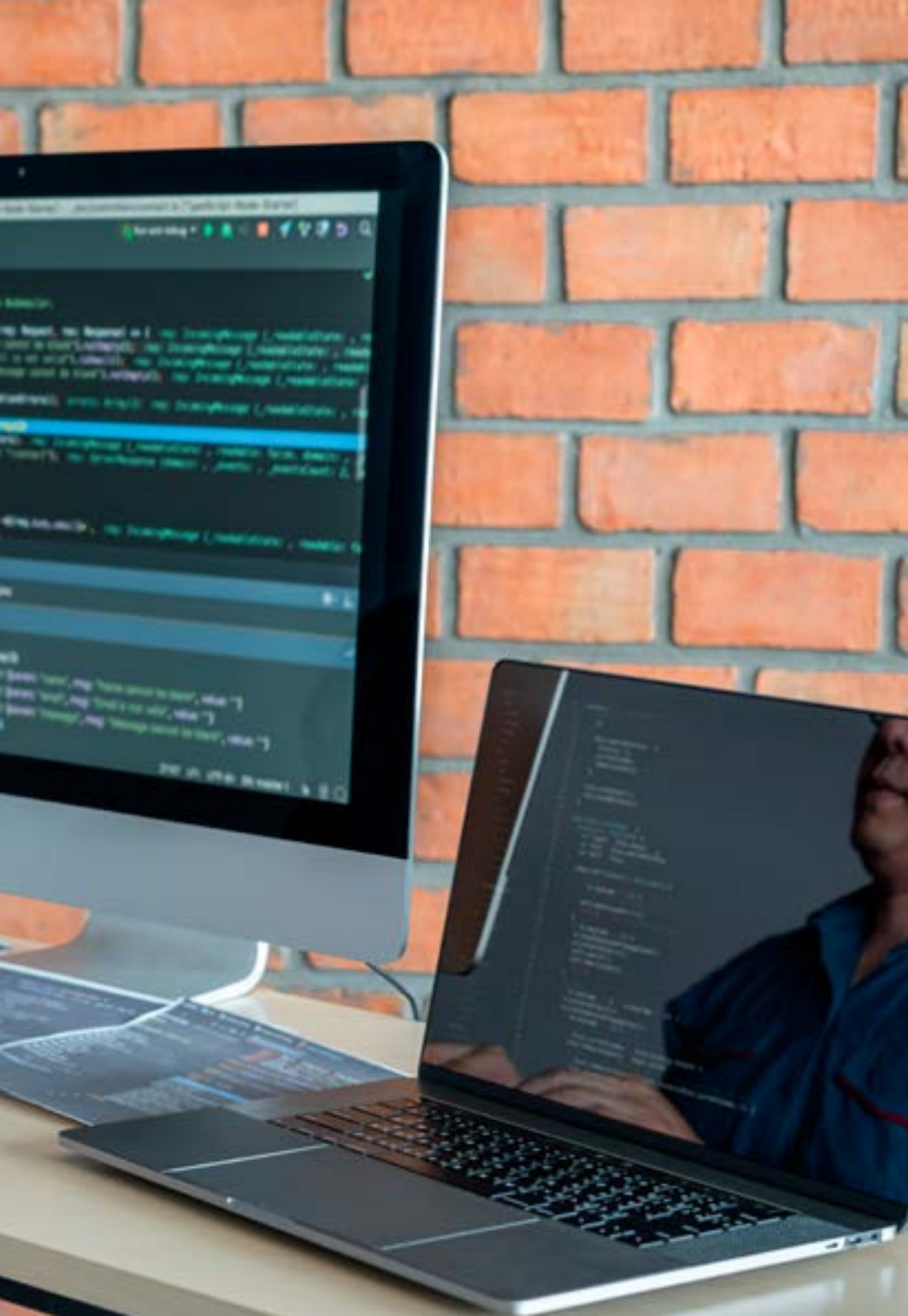
Ce diplôme vous permettra de donner un coup de pouce à votre carrière en acquérant les connaissances pratiques nécessaires pour planifier et gérer des projets faisant appel aux technologies mobiles"



Compétences générales

- ◆ Développer des applications mobiles pour les appareils mobiles Android et iOS
- ◆ Renforcer les *skills* nécessaires à tous les stades du cycle de vie du développement logiciel
- ◆ Analyser différentes méthodologies de programmation pour différents appareils et cas d'utilisation
- ◆ Maîtriser les connaissances spécialisées des différents composants technologiques des appareils mobiles
- ◆ Comprendre les environnements de travail des différents langages et styles de programmation d'applications mobiles
- ◆ Réaliser efficacement des applications du point de vue de l'expérience utilisateur afin de les rendre réellement fonctionnelles et attrayantes
- ◆ Maîtriser les éléments structurels d'un Système Android
- ◆ Analyser les différents *Frameworks* à utiliser par des architectures multiples
- ◆ Définir les propriétés d'une interface du point de vue de l'utilisabilité
- ◆ Développer des connaissances spécialisées sur le langage de programmation Kotlin et le contexte de son émergence
- ◆ Gérer la *Content Provider* pour l'échange et la sécurisation des données
- ◆ Aborder les outils métiers dans Android: leur efficacité, leur utilisabilité
- ◆ Analyser les éléments d'un design responsive
- ◆ Explorer les nouvelles méthodologies et outils centrés sur l'utilisateur
- ◆ Adapter le système de pipeline automatisé à l'écosystème Android





Compétences spécifiques

- ◆ Analyser les processus de développement logiciel du point de vue traditionnel et agile
- ◆ Développer des considérations générales sur les appareils mobiles
- ◆ Maîtriser les concepts clés de la programmation et de l'Internet, le web et son fonctionnement
- ◆ Comparer le modèle de développement natif et le modèle de développement web multiplateforme basé sur les applications hybrides
- ◆ Déterminer comment utiliser les bases de données dans les applications mobiles
- ◆ Publier une application sur le *Play store*
- ◆ Déterminer les différentes étapes d'un cycle d'intégration continue
- ◆ Comprendre les principes de la culture centrée sur l'utilisateur et comment elle crée une nouvelle position pour les professionnels de l'expérience client
- ◆ Résoudre les problèmes de sécurité sur les appareils mobiles
- ◆ Utiliser des ressources alternatives en analysant les dispositions (*layout*) disponibles
- ◆ Développer les différentes formes de gestion du flux de code en Kotlin et ses capacités
- ◆ Améliorer la productivité des applications grâce aux capacités différentielles du langage Kotlin
- ◆ Développer des extensions et des objets compagnons en Kotlin
- ◆ Utiliser des bases de données ou des services de réseau pour créer des fichiers dans différents formats
- ◆ Manipuler la bibliothèque *Room* comme une abstraction pour l'utilisation de SQLite sur Android
- ◆ Évaluer les outils de contrôle des appareils Android, en analysant les possibilités de l'IdO et de la plateforme Android
- ◆ Adopter le *responsive design* pour offrir une expérience utilisateur plus satisfaisante
- ◆ Combiner les techniques de prototypage et *Wireframing* avec les nouvelles technologies de rupture telles que l'intelligence Artificielle et l'Internet des Objets

04

Direction de la formation

Ce Mastère Avancé en Développement d'Applications Mobiles, Expert Android est dirigé par des enseignants ayant une connaissance et une expérience approfondies des nouvelles technologies, de l'architecture de solutions et de l'infrastructure numérique, des experts en programmation Android et des développeurs d'applications. Leurs vastes programmes d'études offrent l'informatique une garantie de la qualité du contenu sélectionné pour cette qualification. L'objectif est d'optimiser le processus d'apprentissage pour les professionnels qui recherchent dans cet espace la contribution dont ils ont besoin pour leur réussite professionnelle.



“

La compilation de toutes les expériences des informaticiens et des ingénieurs spécialisés dans le développement d'applications ajoutera de la valeur à vos propositions et vous rendra plus attractif sur le marché du travail"

Directeur invité international

Colin Lee est un développeur d'applications mobiles à succès, spécialisé dans le **code natif Android**, dont l'influence s'étend à l'échelle internationale. L'expert fait autorité dans la **région des Twin Cities et dans le maniement de Kotlin**. L'une de ses contributions les plus récentes a consisté à démontrer, à l'aide d'un code réel, comment **construire rapidement un navigateur** en utilisant le langage de programmation susmentionné et les composants de navigateur open source de Mozilla pour Android.

En outre, ses applications ont été associées à de grandes entreprises mondiales. Par exemple, il a été chargé de créer des **solutions numériques pour Pearson**, l'un des plus grands éditeurs au monde. Il a également développé un **enregistreur vidéo Android** de bas niveau pour la startup Flipgrid, rachetée par la suite par Microsoft.

Il a également mis au point un VPN Android pour un grand **client du secteur du conseil**. Il est également le créateur d'un outil de gestion du fret mis en œuvre par la transnationale **Amazon** pour faciliter le travail de ses camionneurs contractuels. Il a également participé à la création des **versions mobiles du navigateur Firefox** pour Mozilla.

Aujourd'hui, il travaille en tant qu'entrepreneur, notamment sur des **revues de code** et des **contrôles de sécurité**. Son impact sur le développement d'applications mobiles et l'expérience qu'il a acquise au fil des ans font de lui une figure de proue dans l'arène technologique mondiale.



Dr. Lee, Colin

- Directeur de ColinTheShots LLC
- Ingénieur Logiciel Android pour Specto Inc.
- Ingénieur Android Senior pour Mozilla
- Ingénieur Développeur de Logiciels pour Amazon
- Ingénieur en Applications Mobiles pour Flipgrid
- Spécialiste de la Configuration logicielle pour Pearson VUE
- Licence de l'Université de Floride

“

Grâce à TECH, vous pourrez apprendre avec les meilleurs professionnels du monde”

Direction



M. Olalla Bonal, Martín

- ♦ *Blockchain Technical Specialist* chez IBM SPGI
- ♦ Technicien en Électronique Numérique
- ♦ Architecte *Blockchain*
- ♦ Architecte Infrastructure chez Banca
- ♦ Formation *Hyperledger Fabric* pour les entreprises
- ♦ Formation *Blockchain* orientée business pour les entreprises
- ♦ Gestion de projet et mise en œuvre de solutions
- ♦ Plus de 25 ans d'expérience dans le secteur IT

Professeurs

M. Gozalo Fernández, Juan Luis

- ◆ Ingénieur en informatique
- ◆ Professeur associé aux Experts de l'Université DevOps et au Développement d'Applications *Blockchain* à l'Université Internationale de La Rioja
- ◆ Ancien Blockchain DevOps chez Alastria
- ◆ Responsable de la Technologie des Niveaux de Service chez Santander Espagne
- ◆ Directeur du Développement des Applications Mobiles Tinkerlink chez Cronos Telecom
- ◆ Directeur de la Technologie de Gestion des Services Informatiques à la Barclays Bank Espagne
- ◆ Diplôme en Ingénierie Informatique à l'UNED
- ◆ Spécialisation en Deep Learning en DeepLearning.ai

M. Gómez Rodríguez, Antonio

- ◆ Ingénieur de solutions Cloud chez Oracle
- ◆ Chef de projet dans des groupes d'affaires importants tels que Sopra Group et Everis
- ◆ Chef de projet dans "Entreprise publique de Gestion des Programmes Culturels Conseil de Culture de l'Andalousie
- ◆ Licence en Génie Supérieur des Télécommunications de l'Université Polytechnique de Catalogne
- ◆ Diplôme en Technologies et Systèmes d'Information, Institut Catalan de Technologie
- ◆ E-Business Master, École de Commerce La Salle
- ◆ Analyste des Systèmes d'Information. Sopra Group

M. Guerrero Díaz-Pintado, Arturo

- ◆ Ingénieur réseau R&D chez Telefonica
- ◆ Ingénieur technique avant-vente pour le portefeuille *Watson Customer Engagement* (solutions de Marketing et *Customer Experience*) en Espagne, au Portugal, en Grèce et en Israël chez IBM
- ◆ Consultant en services professionnels travaillant avec des organisations de pointe en Europe, au Moyen-Orient et en Amérique Latine, depuis IBM
- ◆ Diplômé en Ingénierie Supérieure des Télécommunications de l'Université d'Alcalá et de la *Danish Technical University*
- ◆ Plusieurs collaborations exceptionnelles avec des universités et des centres d'enseignement supérieur de renom sur des sujets liés à la technologie tels que l'Intelligence Artificielle, *Internet of Things*, *Cloud*, *Customer Experience* y *Digital Transformation*

M. Villot Guisán, Pablo

- ◆ Architecte Cloud, Exponential Subject Mater Expert *Blockchain* en KPMG
- ◆ Cloud, Exponential Solutions et Subject Mater Expert Blockchain et Architecte d'intégration chez Everis
- ◆ Développeur et responsable technique d'applications web et de bureautique lourde pour le secteur de la logistique commerciale d'Inditex, Connectis
- ◆ Certifié Microsoft en ACEM: *Cloud Platform*
- ◆ Diplômé en Ingénierie Informatique de l'Université de La Coruña

M. Pérez Rico, Javier

- ◆ Responsable Technique Android chez Nologis
- ◆ Responsable technique Android chez Seekle
- ◆ Programmeur Android chez Gowex-Ideup
- ◆ Programmeur Android Junior chez Tecnocom
- ◆ Conférencier au Ile Symposium iTest, E@tic2011
- ◆ Diplômé en Ingénierie en systèmes informatiques techniques à l'Université Complutense de Madrid
- ◆ Maîtrise en recherche à l'Université Complutense de Madrid

M. Noguera Rodríguez, Pablo

- ◆ Développeur Java Expert: JSE, JEE et Android - Ilabora Formation
- ◆ Programmation d'Applications Android - EOI - Madrid
- ◆ Développeur d'Applications Natives (iOS et Android) - Starman Aviation (Aviaze App)
- ◆ Développeur d'Applications Natives (iOS) - Stef (Mtrack App)
- ◆ Développeur d'Applications Natives (iOS et Android) - Bitnovo (Bitnovo App)

M. Gutiérrez Blanco, Víctor

- ◆ Actuel iOS Team Leader en DriveSmart
- ◆ iOS Team Leader en Keytree Spain
- ◆ iOS Leader en SeekleLabs
- ◆ iOS & Python Developer en Web Partners
- ◆ Full-Stack Developer en Let's Health
- ◆ Front-End Lead Developer en Indra y en Intelygenz
- ◆ Diplôme d'Ingénieur en Informatique de l'Universidad Pontificia de Salamanca, Madrid





M. Frias Favero, Pedro Luis

- ◆ CTO - Swearit Technologies
- ◆ COO - Key Identification
- ◆ Full Stack Developer - Ironhack
- ◆ Diplôme d'Ingénieur Industriel de l'Université Yacambú
- ◆ Expert en *Blockchain* et Applications Décentralisées à l'Université d'Alcalá

M. Jiménez Pérez, Carlos

- ◆ Développeur Android senior chez OnTheSpot - Telefónica Tech
- ◆ Professeur Adjoint en Université Carlos III de Madrid
- ◆ Licence en Génie Automatique et Électronique
- ◆ Master en Ingénierie des Systèmes Électroniques et de leurs Applications

05 Structure et contenu

Dans le but de fournir à l'informaticien le meilleur contenu pour le développement réussi d'applications mobiles, TECH dispose d'une équipe de professionnels de la programmation, du développement d'applications et des nouvelles technologies qui ont sélectionné des contenus exclusifs et qui accompagneront l'informaticien dans le processus d'étude au cours des modules à développer en 24 mois. A partir d'une plateforme en ligne, les diplômés peuvent accéder à un contenu interactif, distribué en vidéos, graphiques, exemples basés sur des problèmes, matériel audiovisuel et avec une large communauté pour discuter des questions soulevées.



“

Vous disposerez du meilleur contenu audiovisuel avec des exercices pratiques et des formats interactifs pour rendre votre processus d'apprentissage agile et efficace"

Module 1. Méthodologies de programmation dans le Développement d'Applications Mobiles

- 1.1. Processus de Développement de Software
 - 1.1.1. *Waterfall*
 - 1.1.2. *Spiral*
 - 1.1.3. RUP
 - 1.1.4. V-Model
- 1.2. Processus de développement software agile
 - 1.2.1. *Scrum*
 - 1.2.2. XP
 - 1.2.3. Kanban
- 1.3. Langage de Modélisation Unifié (UML)
 - 1.3.1. UML
 - 1.3.2. Types de modélisation
 - 1.3.3. Blocs de base d'UML
- 1.4. Diagrammes comportementaux 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. Diagramas UML Structurels
 - 1.5.1. *Diagramas UML Structurels*
 - 1.5.2. *Object diagram*
 - 1.5.3. *Component diagram*
 - 1.5.4. *Composite structure diagram*
 - 1.5.5. *Deployment diagram*
- 1.6. Schémas de conception créative
 - 1.6.1. *Singleton*
 - 1.6.2. *Prototype*
 - 1.6.3. *Builder*
 - 1.6.4. *Factory*
 - 1.6.5. *Abstract factory*
- 1.7. Modèles de conception structurels
 - 1.7.1. *Décorateur*
 - 1.7.2. *Façade*
 - 1.7.3. *Adapter*
 - 1.7.4. *Bridge*
 - 1.7.5. *Composite*
 - 1.7.6. *Flyweight*
 - 1.7.7. *Proxy*
- 1.8. Modèles de comportement
 - 1.8.1. *Chain of Responsibility*
 - 1.8.2. *Command*
 - 1.8.3. *Itérateur*
 - 1.8.4. *Médiateur*
 - 1.8.5. *Memento*
 - 1.8.6. *Observer*
 - 1.8.7. *State*
 - 1.8.8. *Strategy*
 - 1.8.9. *Template method*
 - 1.8.10. *Visiteur*
- 1.9. *Testing*
 - 1.9.1. Tests unitaires
 - 1.9.2. Test d'intégration
 - 1.9.3. Techniques boîte blanche
 - 1.9.4. Techniques boîte noire

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

Module 2. Technologies de développement d'applications pour les appareils mobiles

- 2.1. Dispositifs mobiles
 - 2.1.1. Dispositif mobile
 - 2.1.2. Infrastructure d'un appareil mobile
 - 2.1.3. Fabricants de hardware
 - 2.1.4. Développeurs de logiciels
 - 2.1.5. Prestataires de services
 - 2.1.6. Fournisseurs de plateformes
 - 2.1.7. Principales plateformes
- 2.2. Composants physiques des appareils mobiles
 - 2.2.1. Stockage
 - 2.2.1.1. Immutabilité
 - 2.2.1.2. Mutable
 - 2.2.1.3. Temporel
 - 2.2.1.4. Externe
 - 2.2.2. Présentateurs
 - 2.2.2.1. Écrans, haut-parleurs, réponses haptiques
 - 2.2.3. Méthodes d'entrée
 - 2.2.3.1. Boutons/claviers
 - 2.2.3.2. Écrans
 - 2.2.3.3. Microphones
 - 2.2.3.4. Détecteurs de mouvement
 - 2.2.4. Sources d'énergie
 - 2.2.4.1. Sources d'énergie
 - 2.2.4.2. Utilisation adaptative des ressources
 - 2.2.4.3. Programmation efficace
 - 2.2.4.4. Développement durable

- 2.3. Processeurs
 - 2.3.1. Processeur central
 - 2.3.2. Autres processeurs abstraits
 - 2.3.3. Processeurs d'intelligence artificielle
- 2.4. Transmetteurs d'information
 - 2.4.1. Longue portée
 - 2.4.2. Moyenne portée
 - 2.4.3. Courte portée
 - 2.4.4. Ultra courte portée
- 2.5. Capteurs
 - 2.5.1. Interne à l'appareil
 - 2.5.2. Environnement
 - 2.5.3. Médical
- 2.6. Composants logiques
 - 2.6.1. Immuable
 - 2.6.2. Fabricant mutable
 - 2.6.3. Disponible pour l'utilisateur
- 2.7. Catégorisation
 - 2.7.1. Ordinateurs portables
 - 2.7.2. Smartphones
 - 2.7.2.1. Tablette
 - 2.7.2.2. Dispositifs multimédias
 - 2.7.2.3. Plug-ins intelligents
 - 2.7.3. Assistants robotiques
- 2.8. Modes d'opération
 - 2.8.1. Hors ligne
 - 2.8.2. Connecté
 - 2.8.3. Toujours disponible
 - 2.8.4. Point à point

- 2.9. Interactions
 - 2.9.1. Interactions médiatisées par les utilisateurs
 - 2.9.2. Interactions médiatisées par le fournisseur
 - 2.9.3. Interactions médiatisées par des dispositifs
 - 2.9.4. Interactions médiées par l'environnement
- 2.10. Sécurité
 - 2.10.1. Mesures mises en œuvre par le fabricant
 - 2.10.2. Mesures mises en œuvre par les fournisseurs
 - 2.10.3. Sécurité mise en œuvre par l'utilisateur
 - 2.10.4. Confidentialité

Module 3. Outils de travail pour le développement d'applications mobiles

- 3.1. Environnement et outils pour le développement d'applications mobiles
 - 3.1.1. Préparation de l'environnement Mac OS
 - 3.1.2. Préparation de l'environnement Linux
 - 3.1.3. Préparation de l'environnement Windows
- 3.2. Ligne de commande
 - 3.2.1. Ligne de commande
 - 3.2.2. Émulateurs
 - 3.2.3. Interprète de commande
 - 3.2.4. Création de dossiers
 - 3.2.5. Création de dossiers
 - 3.2.6. Navigation
 - 3.2.7. Gestion des fichiers et des dossiers à l'aide de l'interface de ligne de commande
 - 3.2.8. Permissions
 - 3.2.9. SSH
 - 3.2.10. Liste des commandes

- 3.3. Dépôt de Software Git
 - 3.3.1. Système de contrôle des versions
 - 3.3.2. Git
 - 3.3.3. Paramètres
 - 3.3.4. Référentiel
 - 3.3.5. Branches
 - 3.3.6. Gestion des succursales
 - 3.3.7. Flux de travail
 - 3.3.8. Merge
 - 3.3.9. Commandes
- 3.4. Service Web de contrôle des versions
 - 3.4.1. Dépôts à distance
 - 3.4.2. Paramètres
 - 3.4.3. Authentification
 - 3.4.4. Software Fork
 - 3.4.5. Commande Git Clone
 - 3.4.6. Référentiel
 - 3.4.7. *Pages Github*
- 3.5. Outils de Développement Avancés pour les Applications de dispositifs mobiles
 - 3.5.1. *Postman*
 - 3.5.2. *Visual studio code*
 - 3.5.3. Interface graphique pour les bases de données
 - 3.5.4. *Hosting*
 - 3.5.5. Des outils complémentaires pour le développement
- 3.6. Le Web à partir d'un prisme de développement d'applications mobiles
 - 3.6.2. Protocoles
 - 3.6.3. Fournisseur de services internet
 - 3.6.4. Adresses IP
 - 3.6.5. Services de Noms DNS
- 3.7. Programmation dans le cadre du développement d'applications pour les appareils mobiles
 - 3.7.1. Programmation dans le cadre du développement d'applications pour les appareils mobiles
 - 3.7.2. Paradigme de programmation
 - 3.7.3. Langages de programmation
- 3.8. Composants de Développement d'Applications pour les dispositifs mobiles
 - 3.8.1. Variables et constantes
 - 3.8.2. Types
 - 3.8.3. Opérateurs
 - 3.8.4. Déclarations
 - 3.8.5. Boucles
 - 3.8.6. Fonctions et objectifs
- 3.9. Structure des données
 - 3.9.1. Structure des données
 - 3.9.2. Types de structure linéaire
 - 3.9.3. Types de structures fonctionnelles
 - 3.9.4. Types de structures d'arborescence
- 3.10. Algorithmes
 - 3.10.1. Algorithmes en programmation. Diviser pour régner
 - 3.10.2. Algorithmes voraces
 - 3.10.3. Programmation dynamique

Module 4. Développement web multiplateforme pour mobiles

- 4.1. Développement Web multi-plateforme
 - 4.1.1. Développement Web multi-plateforme
 - 4.1.2. Apps Hybrides vs. Apps Natives
 - 4.1.3. Technologies pour créer des applications hybrides
- 4.2. *Progressive Web Apps* (PWA)
 - 4.2.1. *Progressive Web Apps* (PWA)
 - 4.2.2. *Progressive Web Apps* (PWA). Caractéristiques
 - 4.2.3. *Progressive Web Apps* (PWA). Construction
 - 4.2.4. *Progressive Web Apps* (PWA). Limites
- 4.3. *Framework Ionic*
 - 4.3.1. *Framework Ionic*. Analyse
 - 4.3.2. *Framework Ionic*. Caractéristiques
 - 4.3.3. Construire une Application avec *Ionic*
- 4.4. *Frameworks* de développement Web
 - 4.4.1. L'Analyse des *Framework* dans le Développement Web
 - 4.4.2. *Frameworks* de Développement Web
 - 4.4.3. Comparaison des *Frameworks* web
- 4.5. *Framework* Angular
 - 4.5.1. *Framework* Angular
 - 4.5.2. Utilisation d'Angular dans le Développement d'Applications Multiplateformes
 - 4.5.3. Angular + *Ionic*
 - 4.5.4. Créer des Applications en Angular
- 4.6. Bibliothèque de Développement *React*
 - 4.6.1. Bibliothèque *JavaScript React*
 - 4.6.2. Bibliothèque *JavaScript React*. Utilisation
 - 4.6.3. *React Native*
 - 4.6.4. *React* + *Ionic*
 - 4.6.5. Construction de Apps en *React*

- 4.7. *Framework* de Développement Vue
 - 4.7.1. *Framework* de Développement Vue
 - 4.7.2. *Framework* de Développement Vue. Utilisation
 - 4.7.3. Vue + *Ionic*
 - 4.7.4. Créer des applications avec Vue
- 4.8. *Frameworks* de Développement Electron
 - 4.8.1. *Frameworks* de Développement Electron
 - 4.8.2. *Frameworks* de Développement Electron. Utilisation
 - 4.8.3. Déployer nos applications également sur le bureau
- 4.9. Outil de Développement de dispositifs mobiles *Flutter*
 - 4.9.1. Outil de Développement de dispositifs mobiles *Flutter*
 - 4.9.2. Utiliser le SDK de *Flutter*
 - 4.9.3. Créer des Applications avec *Flutter*
- 4.10. Outil de Développement de dispositifs mobiles. Comparaison
 - 4.10.1. Outils pour le Développement d'Applications sur les appareils mobiles
 - 4.10.2. *Flutter* vs. *Ionic*
 - 4.10.3. Choisir le *Stack* le plus appropriée pour la création d'une application

Module 5. Base de données pour le Développement d'Applications mobiles

- 5.1. Bases de Données sur les appareils mobiles
 - 5.1.1. Persistance des données dans le développement d'applications mobiles
 - 5.1.2. Capacités des bases de données pour Application mobiles
 - 5.1.3. SQL. *Structured Query Language*
- 5.2. Choix de la base de données pour les applications mobiles
 - 5.2.1. Analyse des Applications sur les appareils mobiles à partir de bases de données
 - 5.2.2. Catégories de base de données
 - 5.2.3. Panorama de base de données
- 5.3. Développement avec SQLite
 - 5.3.1. Base de données SQLite
 - 5.3.2. Déploiement des modèles
 - 5.3.3. Connexion à SQLite

- 5.4. Développement avec Oracle Berkeley DB
 - 5.4.1. Base de données Berkeley DB
 - 5.4.2. Déploiement du modèle
 - 5.4.3. Connexion à Berkeley DB
- 5.5. Développement avec Realm
 - 5.5.1. Capacités du Realm
 - 5.5.2. Création d'une base de données dans Realm
 - 5.5.3. Se connecter à Realm
- 5.6. Développement avec CouchDB Lite
 - 5.6.1. Base de données CouchDB Lite
 - 5.6.2. Créer une base de données avec CouchDB Lite
 - 5.6.3. Connecter à CouchDB Lite
- 5.7. Développement avec les bases de données centralisées MySQL
 - 5.7.1. Base de Données MySQL
 - 5.7.2. Déploiement de modèle relationnel avec MySQL
 - 5.7.3. Connexion à MySQL
- 5.8. Développements centralisés. Oracle, MS SQL Server, MongoDB
 - 5.8.1. Développement avec Oracle
 - 5.8.2. Développement avec MS SQL Server
 - 5.8.3. Développement avec MongoDB
- 5.9. Données Graphiques
 - 5.9.1. Base de Données orientée graphe
 - 5.9.2. Création de base de données avec Neo4j
 - 5.9.3. Se Connecter à Neo4j depuis des applications mobiles
- 5.10. Environnements avec capacités de stockage
 - 5.10.1. Développements avec Firebase
 - 5.10.2. Développements avec Core Data
 - 5.10.3. Développement avec Visual Builder Cloud Service

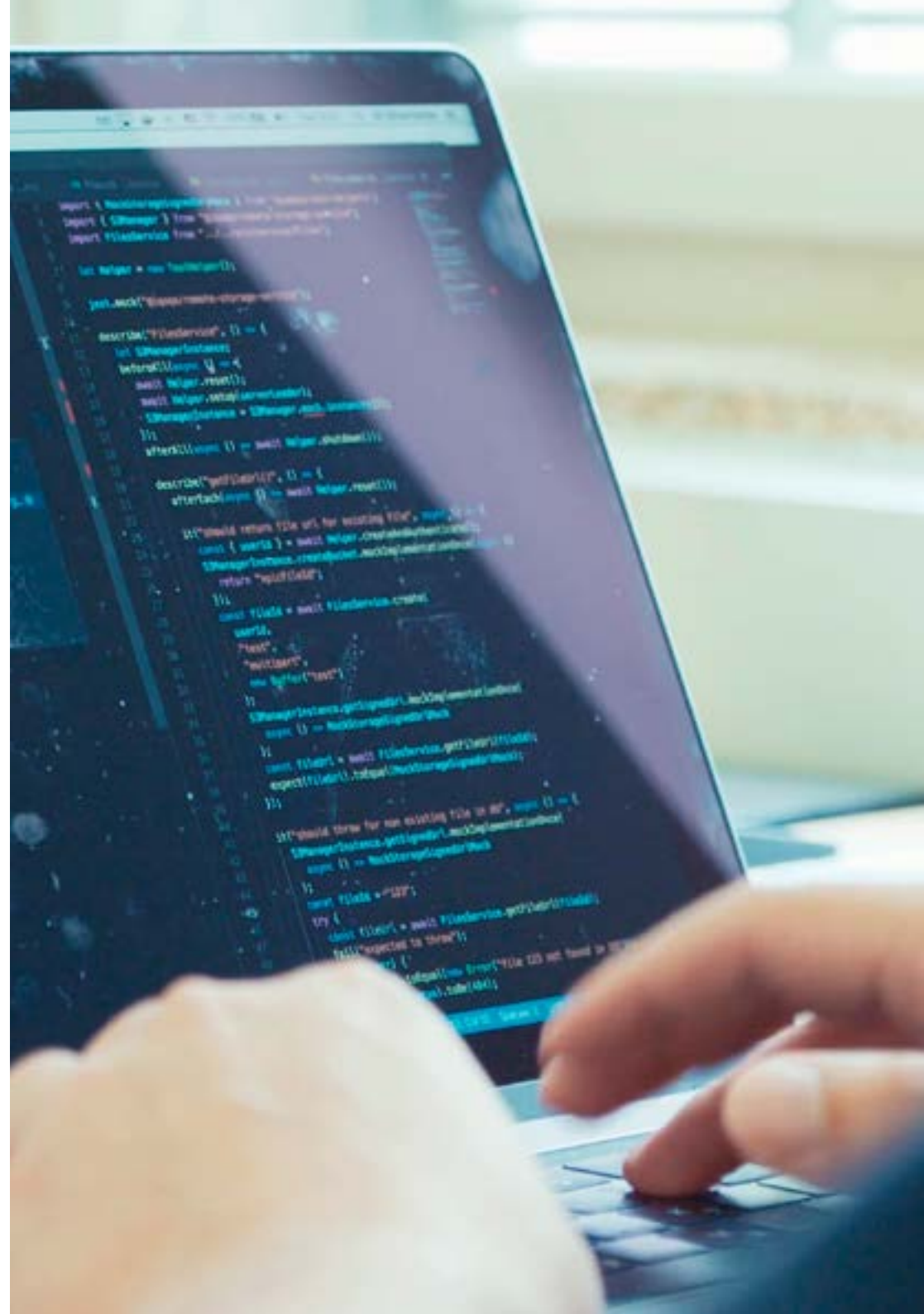
Module 6. Développement d'Applications pour Systèmes iOS

- 6.1. Environnement de Développement Xcode
 - 6.1.1. Création d'un projet
 - 6.1.2. Configurer un Emulateur pour compiler
 - 6.1.3. Configuration d'un téléphone physique à compiler
- 6.2. Langage de programmation Swift
 - 6.2.1. Swift I: langage de programmation
 - 6.2.2. Swift II: fonctions et Loops
 - 6.2.3. Swift III: lambdas et Structs
- 6.3. Bibliothèques et Pods Cocoa
 - 6.3.1. Pods: installation
 - 6.3.2. Configuration de Cocoa Pods
 - 6.3.3. Structure des Cocoa Pods
- 6.4. Bibliothèques: Api, Base de données et R.swift
 - 6.4.1. Alamofire
 - 6.4.2. Bases de données SQL avec GRDB
 - 6.4.3. R.swift
- 6.5. Conception de l'écran
 - 6.5.1. Conception du storyboard
 - 6.5.2. Conception responsive
 - 6.5.3. Conception de vues par code et SwiftUI
- 6.6. Assemblage d'une vue
 - 6.6.1. UIViewController et son cycle de vie
 - 6.6.2. Interaction entre les différents écrans
 - 6.6.3. Types de transitions et modalités
- 6.7. Capteurs et localisation
 - 6.7.1. Accès aux capteurs
 - 6.7.2. Accès à l'emplacement du premier plan
 - 6.7.2. Contexte Localisation Accès

- 6.8. Architectures
 - 6.8.1. MVP
 - 6.8.2. VIPER
 - 6.8.3. Architecture de Développement IOS
- 6.9. Monétisation et *Analytics*
 - 6.9.1. *Analytique Firebase*
 - 6.9.2. *Firebase crashlytics*
 - 6.9.3. Monétisation et annonces avec Google ADMob
- 6.10. App Store et versioning
 - 6.10.1. Configuration d'un compte App Store
 - 6.10.2. Versions d'essai (*test flight*)
 - 6.10.3. Lancement de la production

Module 7. Déploiements d'intégration continue pour mobiles

- 7.1. DevSecOps
 - 7.1.1. DevSecOps. Utilisation
 - 7.1.2. Analyseurs statiques
 - 7.1.3. Analyse dynamique essais de sécurité
- 7.2. Contrôle continu
 - 7.2.1. Contrôle continu
 - 7.2.2. Contrôle continu Analyse et avantages
 - 7.2.3. Contrôle continu Plateformes
- 7.3. Mise en œuvre
 - 7.3.1. Mise en œuvre de la machine
 - 7.3.2. Mise en œuvre de la machine partagée
 - 7.3.3. Mise en œuvre basée sur les services en cloud
 - 7.3.4. Gestion de la configuration



Module 8. *User Experience* dans dispositifs Mobiles

- 8.1. *User experience*
 - 8.1.1. *Client Experience*
 - 8.1.2. *Client Experience*. Exigences
 - 8.1.3. Bidirectionnalité avec le client
- 8.2. *Client Experience*. Objectifs et équipement
 - 8.2.1. *Client Experience*. Objectifs et équipement
 - 8.2.2. Processus itératifs
 - 8.2.3. Informations nécessaires
- 8.3. Micro-interactions
 - 8.3.1. Relation *End-to-end*
 - 8.3.2. Interactions
 - 8.3.3. Omnichannel
- 8.4. Comportement des utilisateurs
 - 8.4.1. Conception des fondations
 - 8.4.2. Analyse du web et des sessions
 - 8.4.3. Experts en analyse
- 8.5. Technologie de pointe
 - 8.5.1. *Machine Learning*
 - 8.5.2. Blockchain
 - 8.5.3. Internet des objets
- 8.6. Composants techniques
 - 8.6.1. Composants techniques
 - 8.6.2. Composants avancés: dispositifs
 - 8.6.3. Composants avancés: différents profils
- 8.7. Utilisabilité
 - 8.7.1. Heuristique de Nielsen
 - 8.7.2. Test utilisateur
 - 8.7.3. Utilisabilité. Erreurs
- 8.8. Techniques d'UX. *User experience*
 - 8.8.1. Règles
 - 8.8.2. *Prototyping*
 - 8.8.3. Outils de *Low-Code*

- 8.9. Stratégie Visuelle
 - 8.9.1. Concepteur de *User Interface*
 - 8.9.2. Travail sur *User Interface* sur le Web
 - 8.9.3. Travail sur *User Interface* dans les Applications
- 8.10. *Developer Frameworks*
 - 8.10.1. *Frameworks* de CX
 - 8.10.2. *Frameworks* de UX
 - 8.10.3. *Frameworks* de UI

Module 9. Sécurité des appareils mobiles

- 9.1. Architecture de sécurité des dispositifs mobiles
 - 9.1.1. Sécurité physique des dispositifs
 - 9.1.2. Sécurité du système d'exploitation
 - 9.1.3. Sécurité des applications
 - 9.1.4. Sécurité des données
 - 9.1.5. Sécurité des communications
 - 9.1.6. Sécurité des dispositifs d'entreprise
- 9.2. Sécurisation du Hardware mobile
 - 9.2.1. Dispositifs mobiles
 - 9.2.2. Dispositifs wearables
 - 9.2.3. Automobile
 - 9.2.4. Dispositifs IOT
 - 9.2.5. Dispositifs TV
- 9.3. Sécurisation des Systèmes d'Exploitation
 - 9.3.1. Appareils mobiles Android
 - 9.3.2. Appareils mobiles Apple IOS
 - 9.3.3. Autres dispositifs mobiles existants: Blackberry, etc.
 - 9.3.4. Dispositifs wearables
 - 9.3.5. Systèmes d'exploitation automobile
 - 9.3.6. Appareils mobiles dans *Internet of Things* (IoT)
 - 9.3.7. Appareils *SmartTV*

- 9.4. Titrisation d'applications mobiles
 - 9.4.1. Appareils mobiles Android
 - 9.4.2. Appareils mobiles Apple IOS
 - 9.4.3. Autres dispositifs mobiles. Blackberry
 - 9.4.4. Dispositifs *Wearables*
 - 9.4.5. Systèmes d'exploitation automobile
 - 9.4.6. Appareils mobiles dans *Internet of Things* (IoT)
 - 9.4.7. Appareils *SmartTV*
- 9.5. Sécurisation des données dans les applications mobiles
 - 9.5.1. Appareils mobiles Android
 - 9.5.2. Appareils mobiles Apple IOS
 - 9.5.3. Autres dispositifs mobiles. Blackberry
 - 9.5.4. Dispositifs *Wearables*
 - 9.5.5. Systèmes d'exploitation automobile
 - 9.5.6. Les Appareils mobiles dans *Internet of Things* (IoT)
 - 9.5.7. Appareils *SmartTV*
- 9.6. Sécurité sur les *Market Places* marchés mobiles
 - 9.6.1. *Google Play* de Google
 - 9.6.2. *Play Store* de Apple
 - 9.6.3. Autres *Market Places*
 - 9.6.4. *Rooting* des appareils mobiles
- 9.7. Solutions de sécurité Multi-plateformes
 - 9.7.1. *Mobile Device Management* (MDM) unique
 - 9.7.2. Types de solutions sur le marché
 - 9.7.3. Sécurisation des dispositifs à l'aide de MDM (*Master Data Management*)
- 9.8. Développement d'applications mobiles sécurisées
 - 9.8.1. Utilisation de modèles pour un développement sécurisé
 - 9.8.2. Gestion intégrée des tests de sécurité
 - 9.8.3. Déploiement sécurisé des applications

- 9.9. Gestion des permissions sur les appareils mobiles
 - 9.9.1. Système de permis
 - 9.9.2. Processus d'exécution dans le noyau
 - 9.9.3. Fils et événements d'exécution
- 9.10. Recommandations de sécurité pour les appareils mobiles
 - 9.10.1. Recommandations de la NSA pour les appareils mobiles
 - 9.10.2. Recommandations de la INCIBE pour les appareils mobiles
 - 9.10.3. ISO 27001:2013 Annexe

Module 10. Langage de programmation Android

- 10.1. Plateforme Android
 - 10.1.1. Plateforme Android
 - 10.1.2. Système d'Exploitation Android
 - 10.1.3. *Open Handset Alliance* dans le développement Android
- 10.2. Architecture Android
 - 10.2.1. Éléments d'architecture d'un Système Android
 - 10.2.2. Communication entre les éléments
 - 10.2.3. Extensibilité de l'architecture Android
 - 10.2.4. Gestion des Ressources Machines: Batterie et Mémoire
 - 10.2.5. Emulateurs Android
- 10.3. Noyau Linux d'Android
 - 10.3.1. Composition du noyau
 - 10.3.2. Éléments structurels du noyau
 - 10.3.3. La Machine Virtuelle Dalvik
 - 10.3.4. La Machine Virtuelle d'exécution Android (ART)
- 10.4. Bibliothèques Natives d'Android
 - 10.4.1. Bibliothèques Natives d'Android
 - 10.4.2. Bibliothèques de compatibilité (*Support Library*)
 - 10.4.3. Bibliothèques natives et extensibilité

- 10.5. Le Système de Fichiers et de Données d'Android
 - 10.5.1. Structure d'une application Android typique
 - 10.5.2. Système de fichiers YAFFS2 et ext4
 - 10.5.3. Utilisation de SQLite et Room pour la gestion des données
- 10.6. Sécurité d'Android
 - 10.6.1. Système de Permissions
 - 10.6.2. Signatures numériques dans le *Android Application Package* (APK)
 - 10.6.3. Processus d'exécution dans le noyau
 - 10.6.4. Fils d'exécution et événements
- 10.7. Composants Structurels d'une application standard
 - 10.7.1. Vue (*View*)
 - 10.7.2. Activité (*Activity*)
 - 10.7.3. Fragment (*Fragment*)
 - 10.7.4. Service (*Service*)
 - 10.7.5. Intention (*Intent*)
 - 10.7.6. *Broadcasts Receiver* y *Content Provider*
 - 10.7.7. Gestion des données et préférences des utilisateurs
- 10.8. Versions d'Android
 - 10.8.1. Versions d'Android
 - 10.8.2. Déploiement des versions d'Android
 - 10.8.3. Déploiement de la distribution d'Android
 - 10.8.4. Android par rapport à Apple IOS et à d'autres systèmes mobiles
- 10.9. Android pour les véhicules
 - 10.9.1. Android et le monde de l'automobile
 - 10.9.2. Éléments structurels d'un système Android pour l'automobile
 - 10.9.3. Communication entre les appareils
- 10.10. Android dans la Domotique, *Wearable* et en *Internet of Things* (IoT)
 - 10.10.1. Le monde connecté
 - 10.10.2. Éléments structurels d'un système Domotique Android
 - 10.10.3. Éléments d'Android *Wearable*
 - 10.10.4. Android dans le *Internet of Things* (IoT)

Module 11. *Frameworks* dans développement d'applications Android

- 11.1. *Frameworks* dans Développement d'Applications Android
 - 11.1.1. *Framework* dans Développement d'Applications Android
 - 11.1.2. *Frameworks*. Typologie
 - 11.1.3. Choix du *Framework* du projet
- 11.2. Mise en œuvre des *Frameworks* sur Android
 - 11.2.1. *Frameworks Core* d'Android pour Java/Kotlin
 - 11.2.2. *Jetpack Composer*
 - 11.2.3. *Frameworks* dans d'autres langues
- 11.3. Systèmes de Gestion de Bibliothèque en Cours de Développement
 - 11.3.1. *Gradle*
 - 11.3.2. Automatisation avec *Gradle*
 - 11.3.3. Outil de Développement Maven
- 11.4. Code propre
 - 11.4.1. Code ordonné
 - 11.4.2. Préparation du Code dans les Applications Android
 - 11.4.3. *Bikeshedding* et priorisation
- 11.5. Modèles de Développement Android
 - 11.5.1. Catégories de patrons
 - 11.5.2. Différences entre les patrons
 - 11.5.3. *Factory*, *Observer* y *Singleton*
- 11.6. MVP Modèle, vue et présentateur
 - 11.6.1. MVC Modèle, vue et contrôleur
 - 11.6.2. Modèle, vue et présentateur
 - 11.6.3. Exemple pratique: Pokémon Battle
- 11.7. MVVM Modèle, vue et *View Model*
 - 11.7.1. MVC vs MVVM
 - 11.7.2. Modèle, vue et *View Model*
 - 11.7.3. Exemple pratique: Pokemon Battle II
- 11.8. *Frameworks* et Bibliothèques les plus utilisés dans Android
 - 11.8.1. Bibliothèques d'interaction avec l'API
 - 11.8.2. Bibliothèques de conversion de données
 - 11.8.3. *Firebase* y *Firebase Analytics*

- 11.9. Framework visual d'Android
 - 11.9.1. Cycle de vie d'une application Android
 - 11.9.2. Conception d'une vue XML
 - 11.9.3. Disposition des éléments et des animations en XML
- 11.10. Frameworks Android dans d'autres langues
 - 11.10.1. *React Native*
 - 11.10.2. *Flutter*
 - 11.10.3. *Ionic*

Module 12. Interfaces et *layouts* dans le développement d'applications Android

- 12.1. Cycle de vie des interfaces Android
 - 12.1.1. Cycle de vie d'Android
 - 12.1.2. Relation processus-activité
 - 12.1.3. Persistance de l'état de l'application
 - 12.1.4. Architecture *Clean* appliquée à Android
- 12.2. Vues dans le Développement d'Applications Android (*Views*)
 - 12.2.1. Couche de Présentation de l'Architecture *Clean*
 - 12.2.2. *Recycler View*
 - 12.2.3. *Adapter View*
- 12.3. Dispositions dans le développement d'applications Android (*Layouts*)
 - 12.3.1. *Layouts* d'Android
 - 12.3.2. *Constraint Layout*
 - 12.3.3. Création de *Layouts* à l'aide de *Android Studio Layout Editor*
- 12.4. Animations dans le développement d'applications Android (*Animations*)
 - 12.4.1. Icônes et images
 - 12.4.2. Transitions
 - 12.4.3. Différence entre l'animation d'une propriété et l'animation d'une vue
- 12.5. Activités et Intentions dans le Développement d'Applications Android (*Activity* et *Intentions*)
 - 12.5.1. Intention explicites et implicites
 - 12.5.2. Barre d'action
 - 12.5.3. Communication entre les activités
- 12.6. Ressources alternatives et ressources du système (*Material Design*, *Cardboard*, etc.)
 - 12.6.1. *Material Design* pour Android
 - 12.6.2. Multimédia sur Android
 - 12.6.3. Réalité virtuelle avec *Google Cardboard for Android NDK*
- 12.7. Styles et thèmes dans le Développement d'Applications Android
 - 12.7.1. Styles dans un projet Android
 - 12.7.2. Thèmes d'un projet Android
 - 12.7.3. Réutilisation des styles et des thèmes
- 12.8. Graphiques, écran tactile et capteurs
 - 12.8.1. Travailler avec des graphiques avancés
 - 12.8.2. Gérer les écrans Tactiles et les Claviers
 - 12.8.3. Utiliser les Capteurs des Appareils Android
- 12.9. Conceptions pour la Réalité Augmentée
 - 12.9.1. Interfaces complexes utilisant la caméra
 - 12.9.2. Capteurs de position et GPS dans la Réalité Augmentée
 - 12.9.3. Présentation sur des écrans non standard
 - 12.9.4. Erreurs et Problèmes courantes
- 12.10. Configuration avancée de l'interface utilisateur avec *AndroidManifest.xml*
 - 12.10.1. La puissance du fichier manifeste Android
 - 12.10.2. Conception programmatique contre conception déclarative
 - 12.10.3. Composants clés de l'archive



Module 13. Langage de Programmation en Applications Android. Kotlin

- 13.1. Langage de Programmation Kotlin
 - 13.1.1. Langage de Programmation Kotlin
 - 13.1.2. Langage de Programmation Kotlin Caractéristiques
 - 13.1.3. Comment fonctionne un programme Kotlin
- 13.2. Programmation en Kotlin
 - 13.2.1. Structure d'un programme en Kotlin
 - 13.2.2. Mots réservés et syntaxe
 - 13.2.3. Écrire dans la console et lire *inputs* de l'utilisateur
- 13.3. Types et variables en Kotlin
 - 13.3.1. Types et variables en Kotlin
 - 13.3.2. Déclarer des variables: Var vs. Val
 - 13.3.3. Opérateurs
 - 13.3.4. Conversions de type
 - 13.3.5. *Tableaux*
- 13.4. Contrôle de Flux Kotlin
 - 13.4.1. Contrôle de Flux
 - 13.4.2. Expressions conditionnelles
 - 13.4.3. Boucles
- 13.5. Fonctions Kotlin
 - 13.5.1. Fonctions Kotlin
 - 13.5.2. Structure d'une Fonction
 - 13.5.3. Fonctions de portée (*Scope functions*)
- 13.6. Classes et Objets en Kotlin
 - 13.6.1. Classes et Objets en Kotlin
 - 13.6.2. Classe
 - 13.6.3. Objets
 - 13.6.4. Constructeurs et initialisation des propriétés
 - 13.6.5. Classes imbriquées et classes intérieures
 - 13.6.6. Classes de données

- 13.7. L'Héritage en Kotlin
 - 13.7.1. Héritage
 - 13.7.2. Superclasses et sous-classes
 - 13.7.3. Écrasement des propriétés et des fonctions
 - 13.7.4. Héritage et autres types de relations entre classes
 - 13.7.5. Classes scellées
 - 13.7.6. Énumération
- 13.8. Classes et interfaces Abstraites en Kotlin
 - 13.8.1. Classes et interfaces abstraites
 - 13.8.2. Classes abstraites
 - 13.8.3. Interfaces
 - 13.8.4. Validation et conversion des opérateurs *is*, *when*, *as*
- 13.9. Collections en Kotlin
 - 13.9.1. Collections en Kotlin
 - 13.9.2. Listes
 - 13.9.3. Sets
 - 13.9.4. Map
- 13.10. Gestion des exceptions et des valeurs nulles en Kotlin
 - 13.10.1. Gestion des Exceptions et des Valeurs Nulles
 - 13.10.2. Valeur nulle, types *nullable* et *non-nullable*
 - 13.10.3. Exceptions

Module 14 Langage de programmation les applications Android. Kotlin avancé. Généricité, programmation fonctionnelle et parallélisme

- 14.1. Généricité en Kotlin
 - 14.1.1. Généricité en Kotlin
 - 14.1.2. Généricité dans les collections, les fonctions, les classes et les interfaces
 - 14.1.3. Covariance et Contravariance: *Out o In*
- 14.2. Fonctions Lambda en Kotlin
 - 14.2.1. Les Fonctions lambda
 - 14.2.2. Structure d'une fonction lambda
 - 14.2.3. Utilisation des fonctions lambda

- 14.3. Fonctions d'ordre supérieur en Kotlin
 - 14.3.1. Fonctions d'ordre supérieur
 - 14.3.2. Fonctions d'ordre supérieur standard en Kotlin
 - 14.3.3. Liaison des appels de fonction
- 14.4. Extensions de Kotlin
 - 14.4.1. Extensions de Kotlin
 - 14.4.2. Fonctions d'extension
 - 14.4.3. Propriétés de l'extension
 - 14.4.4. Objets compagnons
- 14.5. Modèle de *Delegation* en Kotlin
 - 14.5.1. Patron *Delegation*
 - 14.5.2. *Delegation* en Kotlin
 - 14.5.3. Propriétés déléguées
- 14.6. Annotations et Réflexion en Kotlin
 - 14.6.1. Annotations et Réflexion
 - 14.6.2. Annotations en Kotlin
 - 14.6.3. Réflexion Kotlin
- 14.7. *Testing* de Kotlin
 - 14.7.1. *Testing* de Kotlin
 - 14.7.2. *Frameworks* et bibliothèques de *Testing* Kotlin
 - 14.7.3. Kotest
- 14.8. Programmation asynchrone en Kotlin
 - 14.8.1. Programmation asynchrone
 - 14.8.2. Techniques de programmation asynchrone en Kotlin
 - 14.8.3. Comparative Techniques de programmation
- 14.9. Corrutines en Kotlin
 - 14.9.1. Corrutines
 - 14.9.2. Chaînes
 - 14.9.3. Contexte et *Dispatchers*
 - 14.9.4. État partagé et concomitance
 - 14.9.5. Gestion des exceptions dans les corrutines

- 14.10. L'écosystème Kotlin
 - 14.10.1. L'écosystème Kotlin
 - 14.10.2. Bibliothèques Kotlin
 - 14.10.3. Outils pour Kotlin

Module 15. Gestion des Données sur les appareils Android

- 15.1. Gestion des Données Typologie
 - 15.1.1. Gestion des Données sur les appareils mobiles
 - 15.1.2. Alternatives pour la Gestion des Données sur les appareils Android
 - 15.1.3. Génération de données pour l'intelligence Artificielle et Analyse de l'Usage
 - 15.1.4. Outils de mesure de la performance pour une gestion optimale des données
- 15.2. Gestion des Préférences d'utilisateurs
 - 15.2.1. Types de données impliquées dans les fichiers de préférences
 - 15.2.2. Gestion des préférences des utilisateurs
 - 15.2.3. Exportation des préférences. Gestion des permis
- 15.3. Système de stockage de fichiers
 - 15.3.1. Classification des systèmes de fichiers sur les appareils mobiles
 - 15.3.2. Système Interne de fichiers
 - 15.3.3. Système Externe de fichiers
- 15.4. Fichiers JSON comme moyen de stockage sur Android
 - 15.4.1. Informations non structurées dans les fichiers JSON
 - 15.4.2. Librairies de Gestion de Données JSON
 - 15.4.3. Utilisation de JSON dans Android Recommandations et optimisations
- 15.5. Fichiers XML comme stockage sur Android
 - 15.5.1. Format XML dans Android
 - 15.5.2. XML dans les librairies SAX
 - 15.5.3. XML dans les librairies DOM
- 15.6. Base de Données SQLite
 - 15.6.1. Base de données relationnelle pour la Gestion des Données
 - 15.6.2. Utilisation de la base de données
 - 15.6.3. Méthodes SQLite pour la gestion des données

- 15.7. Utilisation avancée des bases de données SQLite
 - 15.7.1. Récupération des défaillances à l'aide de transactions SQLite
 - 15.7.2. Utilisation de la mise en cache pour accélérer l'accès aux données
 - 15.7.3. Bases de données sur les mobiles
- 15.8. Librairie Room
 - 15.8.1. Architecture de la librairie Room
 - 15.8.2. Librairie Room. Fonctionnalité
 - 15.8.3. Librairie Room Avantages et inconvénients
- 15.9. *Content Provider* pour partager des informations
 - 15.9.1. *Content Provider* pour partager des informations
 - 15.9.2. *Content Provider* dans Android. Technicien d'utilisation
 - 15.9.3. Sécurité du *Content Provider*
- 15.10. Collecte de Données dans le Cloud d'Internet
 - 15.10.1. Android et Systèmes de stockage en cloud (*Cloud*)
 - 15.10.2. Services SOAP et REST pour Android
 - 15.10.3. Questions relatives aux Systèmes distribués
 - 15.10.4. Internet comme sauvegarde des données d'application

Module 16. Outils pour les dispositifs Android

- 16.1. Gestion: outils "TO DO"
 - 16.1.1. Outils de Marché
 - 16.1.2. Outils de Marché Comparaison des fonctionnalités
 - 16.1.3. Outils de gestion Différences
- 16.2. MDM: Gestion des appareils d'entreprise
 - 16.2.1. Contrôle des dispositifs d'entreprise
 - 16.2.2. Analyse des principaux outils de marché
 - 16.2.3. Choix de l'outil
- 16.3. CRM: outils de marché
 - 16.3.1. Analyse des outils de marché avec l'application Android
 - 16.3.2. Outils de Marché Efficacité
 - 16.3.3. Outils de Marché Utilisations

- 16.4. Drones avec Android
 - 16.4.1. Applications Android pour le contrôle des drones
 - 16.4.2. Contrôles autonomes
 - 16.4.3. Utilisation de drones sur Android
- 16.5. Android, valeur ajoutée aux plateformes Bancaires
 - 16.5.1. Android sur les plateformes Bancaires
 - 16.5.2. Risques et fraudes des cybercriminels
 - 16.5.3. Utilisations des appareils mobiles
- 16.6. Sécurité des appareils mobiles
 - 16.6.1. Outils de communication et leur Utilisation
 - 16.6.2. Comparaison des outils
 - 16.6.3. Choix de l'outil pour chaque utilisation
- 16.7. Outils de divertissement et de formation
 - 16.7.1. Utilisations
 - 16.7.2. Outils de marché
 - 16.7.3. Comparaison des fonctionnalités des outils de développement Android
- 16.8. IoT Android
 - 16.8.1. Framework Plateformes du marché
 - 16.8.2. Risques et considérations liés à l'IdO Android
 - 16.8.3. Utilisations de IoT dans Android
- 16.9. Efficacité des processus
 - 16.9.1. Analyse des outils du marché pour la création d'applications
 - 16.9.2. Comparaison des outils de création d'applications pour Android
 - 16.9.3. Use Case
- 16.10. Applications les plus téléchargées aujourd'hui
 - 16.10.1. Outils les plus téléchargés aujourd'hui
 - 16.10.2. Regroupement par familles
 - 16.10.3. Principales utilisations, utilisations secondaires et comparaisons avec IOS

Module 17. Conception responsive dans Android

- 17.1. Conception responsive
 - 17.1.1. Conception responsive
 - 17.1.2. Utilisabilité, accessibilité et UX
 - 17.1.3. Conception responsive Avantages et inconvénients
- 17.2. Mobile vs Tablette vs Web vs Smartwatches
 - 17.2.1. Différents formats, différentes tailles, différents besoins
 - 17.2.2. Problèmes en cours de conception
 - 17.2.3. Adaptation et responsive
- 17.3. Guide de style
 - 17.3.1. Guide de style. Utilité
 - 17.3.2. Matériel design
 - 17.3.3. Guide de style propre
- 17.4. Layouting flexible
 - 17.4.1. Layouting Flexible
 - 17.4.2. Layouting Basic
 - 17.4.3. Layouting en Grid
 - 17.4.4. Layouting avec Relative Layout
 - 17.4.5. Layouting avec Constraint Layout
- 17.5. Ressources flexibles
 - 17.5.1. Ressources flexibles
 - 17.5.2. Images
 - 17.5.3. 9 Patch
 - 17.5.4. Ressources globales
- 17.6. Navigation flexible
 - 17.6.1. Navigation flexible
 - 17.6.2. Navigation avec Activities
 - 17.6.3. Navigation avec Fragments

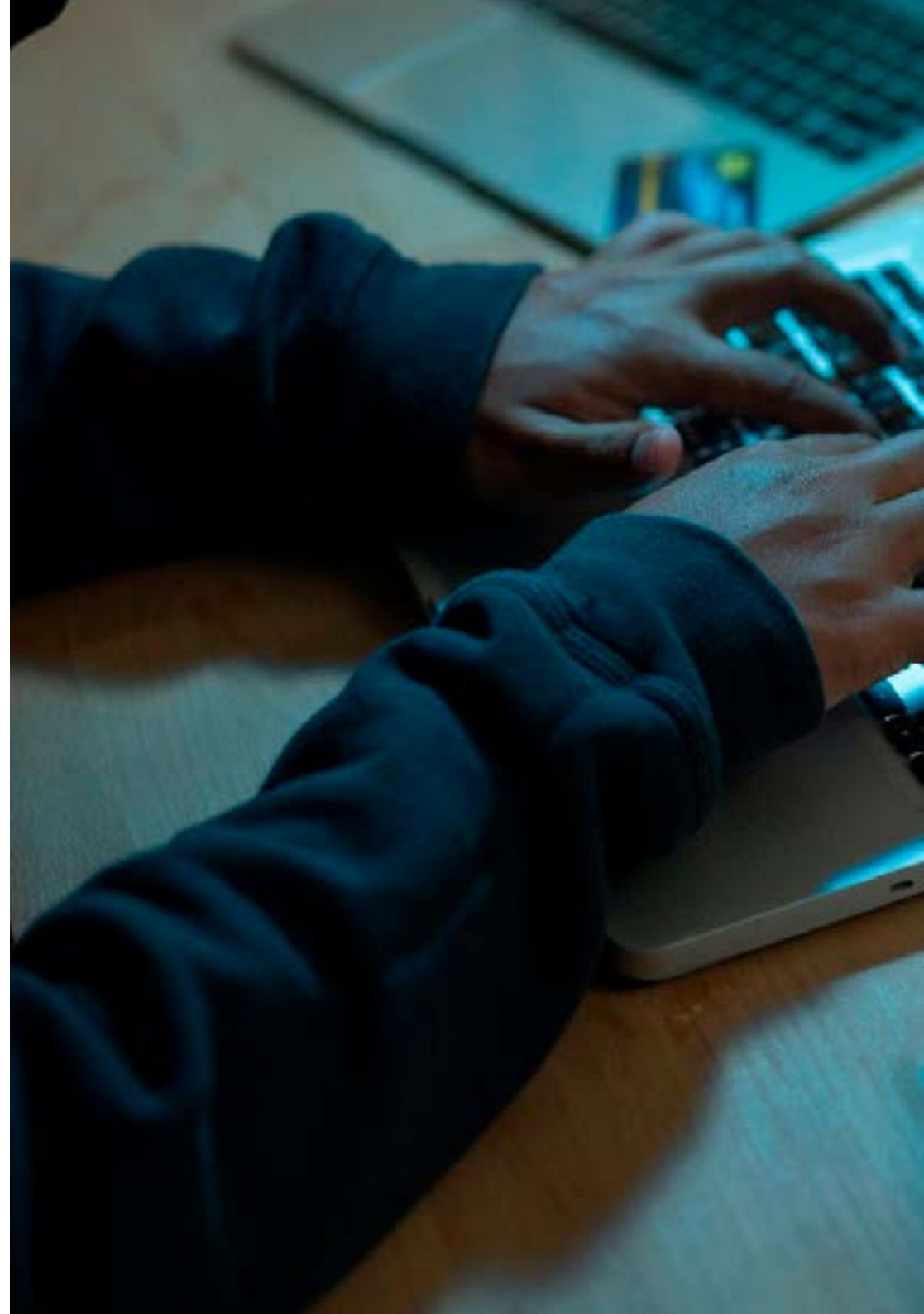
- 17.7. Outils Externes
 - 17.7.1. Générateurs automatiques
 - 17.7.2. Outils de prototypage
 - 17.7.3. Outils de design
- 17.8. *Debug et Tests*
 - 17.8.1. *Debug layouts*
 - 17.8.2. *Tests automatiques*
 - 17.8.3. Développement basé sur les composants
 - 17.8.4. onnes pratiques en matière de *Testing* d'essais
- 17.9. Alternatives au système Android natif I, *Web Pages*
 - 17.9.1. Concevoir dans une *WebView*
 - 17.9.2. *ChromeCustomTabs*
 - 17.9.3. *Debug et Tests* en sites web
- 17.10. Alternatives à la version Native d'Android II. Applications Hybrides
 - 17.10.1. *React/React native*
 - 17.10.2. *Flutter*
 - 17.10.3. *Ionic*
 - 17.10.4. Apache Cordova

Module 18. Marketing des applications Android

- 18.1. De *Customer Service* a *Customer Experience*
 - 18.1.1. *Customer Service*. Développement de la clientèle actuelle
 - 18.1.2. Droit d'accès à l'information. Exigences et besoins
 - 18.1.3. Le rétro-alimentation en tant que source de connaissances
- 18.2. *Customer Journey*
 - 18.2.1. Chemin de l'utilisateur vers la conversion
 - 18.2.2. Micro-segmentation
 - 18.2.3. Expérience à travers les canaux
- 18.3. Mesure de l'expérience Utilisateur
 - 18.3.1. Architecture web et mobiles
 - 18.3.2. L'analyse de session, une nouvelle norme
 - 18.3.3. État de l'Art de l'expérience utilisateur
- 18.4. Marketing des Applications Android
 - 18.4.1. CX+IA
 - 18.4.2. CX+Blockchain
 - 18.4.3. CX+IoT
- 18.5. Produits CX (expérience client)
 - 18.5.1. Normes de l'industrie
 - 18.5.2. Téléprésence
 - 18.5.3. Expérience client pour tous les Acteurs du Développement
- 18.6. Travail centrée sur l'utilisateur
 - 18.6.1. Équipements
 - 18.6.2. Penser en termes de conception
 - 18.6.3. Travail sur le terrain
- 18.7. La Science sur l'utilisateur
 - 18.7.1. La Science sur l'utilisateur Règles d'or
 - 18.7.2. Itération
 - 18.7.3. Erreurs courantes
- 18.8. Prototypage et *Wireframing*
 - 18.8.1. Prototypage et *Wireframing*
 - 18.8.2. *Hands-on*
 - 18.8.3. Niveau avancé
- 18.9. Interfaces mobiles
 - 18.9.1. Conception visuelle Règles
 - 18.9.2. Interface de l'application Clés
 - 18.9.3. Bonnes pratiques en matière de développement d'interface mobile
- 18.10. Bonnes pratiques en matière d'Expérience des Utilisateurs Conseils pour les Développeurs
 - 18.10.1. Niveau Un. Bonnes pratiques en matière de CX
 - 18.10.2. Niveau Deux. Bonnes pratiques en matière de UX
 - 18.10.3. Niveau trois. Bonnes pratiques en matière de UI

Module 19. Cycle de vie d'une application Android Cloud, Playstore et versions

- 19.1. Cycle de vie d'un *software*
 - 19.1.1. Cycle de vie d'un *software*
 - 19.1.2. Méthodologies Agiles
 - 19.1.3. Le cycle continu du *logiciel* agile
- 19.2. Développement manuel de produits
 - 19.2.1. Intégration manuelle
 - 19.2.2. Livraison manuelle
 - 19.2.3. Déploiement manuel
- 19.3. Intégration supervisée
 - 19.3.1. Intégration continue
 - 19.3.2. Examen manuel
 - 19.3.3. Contrôles automatiques statiques
- 19.4. Tests logiques
 - 19.4.1. Tests unitaires
 - 19.4.2. Test d'intégration
 - 19.4.3. Tests comportementaux
- 19.5. Intégration continue
 - 19.5.1. Cycle d'intégration continue
 - 19.5.2. Dépendances entre les intégrations
 - 19.5.3. Intégration continue comme Méthodologie de Gestion des Référentiels
- 19.6. Livraison continue
 - 19.6.1. Livraison continue. Typologie des problèmes à résoudre
 - 19.6.2. Livraison continue. Résolution de problèmes
 - 19.6.3. Avantages de la livraison Continue



- 19.7. Intégration Continue
 - 19.7.1. Déploiement Continue. Typologie des Problèmes à résoudre
 - 19.7.2. Déploiement continu. Résolution de problèmes
- 19.8. *Firebase Test Lab*
 - 19.8.1. Configuration à partir de *GCloud*
 - 19.8.2. Configuration de Jenkins
 - 19.8.3. Utilisation de Jenkins. Avantages
- 19.9. Configuration de la *Gradle*
 - 19.9.1. Système d'Automatisation *Gradle*
 - 19.9.2. Composante de *Gradle Build Flavors*
 - 19.9.3. Composante de *Gradle Linteo*
- 19.10. Cycle de vie d'une Application Android. Exemple
 - 19.10.1. Configuration de *SemaphoreCI* et *GitHub*
 - 19.10.2. Configuration des blocs de travail
 - 19.10.3. Promotions et *Deployment*



Le chemin vers un avenir meilleur commence maintenant et vous en ferez partie. Commencez à concevoir les meilleures applications mobiles avec ce Mastère Avancé"

06

Méthodologie

Ce programme de formation offre une manière différente d'apprendre. Notre méthodologie est développée à travers un mode d'apprentissage cyclique: ***le Relearning***.

Ce système d'enseignement est utilisé, par exemple, dans les écoles de médecine les plus prestigieuses du monde et a été considéré comme l'un des plus efficaces par des publications de premier plan telles que le ***New England Journal of Medicine***.



“

Découvrez Relearning, un système qui renonce à l'apprentissage linéaire conventionnel pour vous emmener à travers des systèmes d'enseignement cycliques: une façon d'apprendre qui s'est avérée extrêmement efficace, en particulier dans les matières qui exigent la mémorisation”

Étude de Cas pour mettre en contexte tout le contenu

Notre programme offre une méthode révolutionnaire de développement des compétences et des connaissances. Notre objectif est de renforcer les compétences dans un contexte changeant, compétitif et hautement exigeant.

“

Avec TECH, vous pouvez expérimenter une manière d'apprendre qui ébranle les fondations des universités traditionnelles du monde entier”



Vous bénéficierez d'un système d'apprentissage basé sur la répétition, avec un enseignement naturel et progressif sur l'ensemble du cursus.



L'étudiant apprendra, par des activités collaboratives et des cas réels, à résoudre des situations complexes dans des environnements commerciaux réels.

Une méthode d'apprentissage innovante et différente

Cette formation TECH est un programme d'enseignement intensif, créé de toutes pièces, qui propose les défis et les décisions les plus exigeants dans ce domaine, tant au niveau national qu'international. Grâce à cette méthodologie, l'épanouissement personnel et professionnel est stimulé, faisant ainsi un pas décisif vers la réussite. La méthode des cas, technique qui constitue la base de ce contenu, permet de suivre la réalité économique, sociale et professionnelle la plus actuelle.

“ Notre programme vous prépare à relever de nouveaux défis dans des environnements incertains et à réussir votre carrière ”

La méthode des cas est le système d'apprentissage le plus largement utilisé dans les meilleures écoles d'informatique du monde depuis qu'elles existent. Développée en 1912 pour que les étudiants en Droit n'apprennent pas seulement le droit sur la base d'un contenu théorique, la méthode des cas consiste à leur présenter des situations réelles complexes afin qu'ils prennent des décisions éclairées et des jugements de valeur sur la manière de les résoudre. En 1924, elle a été établie comme méthode d'enseignement standard à Harvard.

Dans une situation donnée, que doit faire un professionnel? C'est la question à laquelle nous sommes confrontés dans la méthode des cas, une méthode d'apprentissage orientée vers l'action. Tout au long du programme, les étudiants seront confrontés à de multiples cas réels. Ils devront intégrer toutes leurs connaissances, faire des recherches, argumenter et défendre leurs idées et leurs décisions.

Relearning Methodology

TECH combine efficacement la méthodologie des Études de Cas avec un système d'apprentissage 100% en ligne basé sur la répétition, qui associe différents éléments didactiques dans chaque leçon.

Nous enrichissons l'Étude de Cas avec la meilleure méthode d'enseignement 100% en ligne: le Relearning.

En 2019, nous avons obtenu les meilleurs résultats d'apprentissage de toutes les universités en ligne du monde.

À TECH, vous apprendrez avec une méthodologie de pointe conçue pour former les managers du futur. Cette méthode, à la pointe de la pédagogie mondiale, est appelée Relearning.

Notre université est la seule université autorisée à utiliser cette méthode qui a fait ses preuves. En 2019, nous avons réussi à améliorer les niveaux de satisfaction globale de nos étudiants (qualité de l'enseignement, qualité des supports, structure des cours, objectifs...) par rapport aux indicateurs de la meilleure université en ligne.



Dans notre programme, l'apprentissage n'est pas un processus linéaire, mais se déroule en spirale (apprendre, désapprendre, oublier et réapprendre). Par conséquent, chacun de ces éléments est combiné de manière concentrique. Cette méthodologie a permis de former plus de 650.000 diplômés universitaires avec un succès sans précédent dans des domaines aussi divers que la biochimie, la génétique, la chirurgie, le droit international, les compétences en gestion, les sciences du sport, la philosophie, le droit, l'ingénierie, le journalisme, l'histoire, les marchés financiers et les instruments. Tout cela dans un environnement très exigeant, avec un corps étudiant universitaire au profil socio-économique élevé et dont l'âge moyen est de 43.5 ans.

Le Relearning vous permettra d'apprendre avec moins d'efforts et plus de performance, en vous impliquant davantage dans votre formation, en développant un esprit critique, en défendant des arguments et en contrastant les opinions: une équation directe vers le succès.

À partir des dernières preuves scientifiques dans le domaine des neurosciences, non seulement nous savons comment organiser les informations, les idées, les images et les souvenirs, mais nous savons aussi que le lieu et le contexte dans lesquels nous avons appris quelque chose sont fondamentaux pour notre capacité à nous en souvenir et à le stocker dans l'hippocampe, pour le conserver dans notre mémoire à long terme.

De cette manière, et dans ce que l'on appelle Neurocognitive context-dependent e-learning, les différents éléments de notre programme sont reliés au contexte dans lequel le participant développe sa pratique professionnelle.



Ce programme offre le support matériel pédagogique, soigneusement préparé pour les professionnels:



Support d'étude

Tous les contenus didactiques sont créés par les spécialistes qui enseigneront le cours, spécifiquement pour le cours, afin que le développement didactique soit vraiment spécifique et concret.

Ces contenus sont ensuite appliqués au format audiovisuel, pour créer la méthode de travail TECH en ligne. Tout cela, avec les dernières techniques qui offrent des pièces de haute qualité dans chacun des matériaux qui sont mis à la disposition de l'étudiant.



Cours magistraux

Il existe des preuves scientifiques de l'utilité de l'observation par un tiers expert.

La méthode "Learning from an Expert" renforce les connaissances et la mémoire, et donne confiance dans les futures décisions difficiles.



Pratiques en compétences et aptitudes

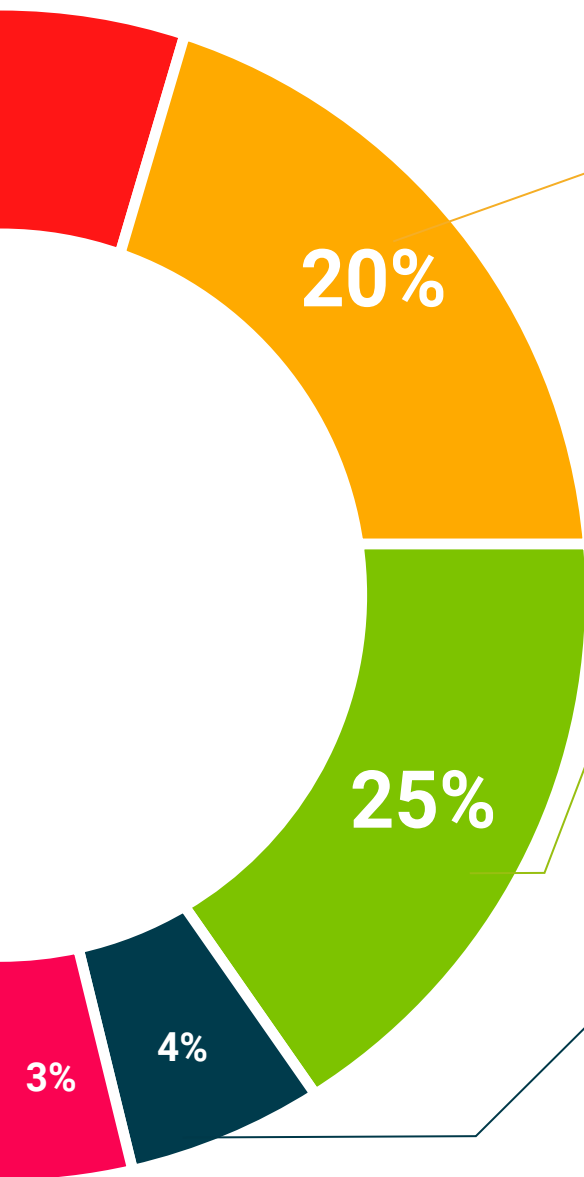
Les étudiants réaliseront des activités visant à développer des compétences et des aptitudes spécifiques dans chaque domaine. Des activités pratiques et dynamiques pour acquérir et développer les compétences et aptitudes qu'un spécialiste doit développer dans le cadre de la mondialisation dans laquelle nous vivons.



Lectures complémentaires

Articles récents, documents de consensus et directives internationales, entre autres. Dans la bibliothèque virtuelle de TECH, l'étudiant aura accès à tout ce dont il a besoin pour compléter sa formation.





Case studies

Ils réaliseront une sélection des meilleures études de cas choisies spécifiquement pour ce diplôme. Des cas présentés, analysés et tutorés par les meilleurs spécialistes de la scène internationale.



Résumés interactifs

L'équipe TECH présente les contenus de manière attrayante et dynamique dans des pilules multimédia comprenant des audios, des vidéos, des images, des diagrammes et des cartes conceptuelles afin de renforcer les connaissances. Ce système éducatif unique pour la présentation de contenu multimédia a été récompensé par Microsoft en tant que "European Success Story".



Testing & Retesting

Les connaissances de l'étudiant sont périodiquement évaluées et réévaluées tout au long du programme, par le biais d'activités et d'exercices d'évaluation et d'auto-évaluation, afin que l'étudiant puisse vérifier comment il atteint ses objectifs.



07 Diplôme

Le Mastère Avancé en Développement d'Applications Mobiles, Expert Android vous garantit, en plus de la formation la plus rigoureuse et la plus actuelle, l'accès à un diplôme universitaire de Mastère Avancé délivré par TECH Université Technologique.



“

Terminez ce programme avec succès et recevez votre diplôme universitaire sans avoir à vous soucier des déplacements ou des formalités administratives”

Ce **Mastère Avancé en Développement d'Applications Mobiles, Expert Android** contient le programme le plus complet et le plus à jour du marché.

Après avoir réussi l'évaluation, l'étudiant recevra par courrier postal* avec accusé de réception son correspondant diplôme de **Mastère Avancé** délivré par **TECH Université Technologique**.

Le diplôme délivré par **TECH Université Technologique** indiquera la note obtenue lors du Mastère Avancé, et répond aux exigences communément demandées par les bourses d'emploi, les concours et les commissions d'évaluation des carrières professionnelles.

Diplôme: **Mastère Avancé en Développement d'Applications Mobiles, Expert Android**
N° d'heures officielles: **3.000 h.**



*Si l'étudiant souhaite que son diplôme version papier possède l'Apostille de La Haye, TECH EDUCATION fera les démarches nécessaires pour son obtention moyennant un coût supplémentaire.

future
santé confiance personnes
éducation information tuteurs
garantie accréditation enseignement
institutions technologie apprentissage
communauté engagement
service personnalisé innovation
connaissance présent qualité
en ligne formation
développement institutions
classe virtuelle langage

tech université
technologique

Mastère Avancé
Développement d'Applications
Mobiles, Expert Android

- » Modalité: en ligne
- » Durée: 2 ans
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

Mastère Avancé

Développement d'Applications
Mobiles, Expert Android