

# Mastère Spécialisé

## Qualité du Software





## Mastère Spécialisé Qualité du Software

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Global University
- » Accréditation: 60 ECTS
- » Horaire: à votre rythme
- » Examens: en ligne

Accès au site web : [www.techstitute.com/fr/information-technology/master/master-qualite-software](http://www.techstitute.com/fr/information-technology/master/master-qualite-software)



# Sommaire

<b>01</b>	<b>Présentation du programme</b>	<i>page 4</i>	<b>02</b>	<b>Pourquoi étudier à TECH?</b>	<i>page 8</i>	<b>03</b>	<b>Programme d'études</b>	<i>page 12</i>
<b>04</b>	<b>Objectifs pédagogiques</b>	<i>page 24</i>	<b>05</b>	<b>Opportunités de carrière</b>	<i>page 30</i>	<b>06</b>	<b>Licences de logiciels incluses</b>	<i>page 34</i>
<b>07</b>	<b>Méthodologie d'étude</b>	<i>page 38</i>	<b>08</b>	<b>Corps enseignant</b>	<i>page 48</i>	<b>09</b>	<b>Diplôme</b>	<i>page 56</i>

01

# Présentation du programme

La Qualité du Software s'est imposée comme un facteur stratégique dans le développement de solutions informatiques modernes. Au-delà de leur fonctionnalité, les systèmes doivent être fiables, efficaces, faciles à entretenir et sûrs. Dans ce contexte, il devient nécessaire de définir des indicateurs objectifs permettant d'évaluer de manière systématique le niveau de qualité atteint dans chaque projet. C'est dans cette optique que TECH a créé un diplôme universitaire pionnier axé sur les modèles les plus innovants pour garantir la Qualité du Software tout au long de son cycle de vie. Il s'agit d'un programme 100 % en ligne, pratique et adapté à l'emploi du temps des professionnels très occupés.





66

Grâce à ce Mastère Spécialisé 100 % en ligne, vous concevez des structures logicielles fortement axées sur la Qualité, la maintenance et l'évolutivité”

Le concept de Qualité du Software a évolué d'une approche purement fonctionnelle vers une vision holistique du produit. Actuellement, il est compris comme un ensemble d'attributs qui déterminent sa capacité à satisfaire des besoins explicites et implicites dans des conditions spécifiques. Face à cela, les spécialistes doivent disposer d'une connaissance approfondie des méthodologies les plus modernes pour évaluer le fonctionnement des architectures distribuées et des environnements DevOps.

Dans ce contexte, TECH présente un Mastère Spécialisé de pointe en Qualité du Software. Le programme d'études approfondira des questions telles que la normalisation des bases de données et le découplage des composants. De même, le programme fournira aux étudiants les clés pour concevoir des architectures évolutives. Les supports pédagogiques approfondiront également l'utilisation des métriques pour évaluer la Qualité des solutions. Grâce à cela, les diplômés développeront une vision globale des processus d'assurance Qualité, maîtrisant tout, de la planification des tests automatisés à la mise en œuvre des normes internationales.

Pour consolider tous ces contenus, TECH utilise la méthode avant-gardiste du *Relearning*, qui consiste à réitérer progressivement les concepts clés pour qu'ils soient correctement assimilés. De plus, la formation universitaire offre aux professionnels une diversité de cas pratiques réels, permettant ainsi aux étudiants de s'exercer dans des environnements simulés afin de les rapprocher de la réalité de la pratique informatique. À cet égard, pour accéder aux ressources éducatives, les diplômés n'auront besoin que d'un appareil électronique capable de se connecter à Internet. De plus, le programme bénéficiera de la collaboration d'un éminent Directeur Invité International, qui dispensera 10 Masterclasses exhaustives.

Ce **Mastère Spécialisé en Qualité du Software** contient le programme le plus complet et le plus actualisé du marché. Ses caractéristiques sont les suivantes :

- Le développement de cas pratiques présentés par des experts en Développement de Software
- Les contenus graphiques, schématiques et éminemment pratiques avec lesquels ils sont conçus fournissent des informations scientifiques et sanitaires essentielles à la pratique professionnelle
- Exercices pratiques permettant de réaliser le processus d'auto-évaluation afin d'améliorer l'apprentissage
- Il met l'accent sur les méthodologies innovantes
- Cours théoriques, questions à l'expert et travail de réflexion individuel
- La possibilité d'accéder aux contenus depuis n'importe quel appareil fixe ou portable doté d'une connexion internet

“

*Un Directeur Invité International prestigieux proposera 10 Masterclasses intensives sur les dernières tendances en matière de Qualité du Software”*

“

*Un programme d'études fondé sur la méthodologie révolutionnaire du Relearning, qui vous permettra d'assimiler efficacement et rapidement des concepts complexes”*

Le corps enseignant comprend des professionnels du domaine du développement de Software, qui apportent à ce programme leur expérience professionnelle, ainsi que des spécialistes reconnus issus d'entreprises de référence 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 académique. Pour ce faire, le professionnel aura l'aide d'un système vidéo interactif innovant créé par des experts reconnus.

*Vous approfondirez votre connaissance des outils modernes d'automatisation pour effectuer des tests fonctionnels, de performance et de régression.*

*Vous comprenez les principes fondamentaux de la Qualité du Software dans le développement de systèmes informatiques avancés.*



02

# Pourquoi étudier à TECH?

TECH est la plus grande Université numérique du monde. Avec un catalogue impressionnant de plus de 14 000 programmes universitaires, disponibles en 11 langues, elle se positionne comme un leader en matière d'employabilité, avec un taux de placement de 99 %. En outre, elle dispose d'un vaste corps professoral composé de plus de 6 000 professeurs de renommée internationale.



66

Étudiez dans la plus grande université numérique  
du monde et assurez votre réussite professionnelle.  
L'avenir commence à TECH™





03

# Programme d'études

Ce programme aborde, dans une perspective critique et interdisciplinaire, le rôle des organismes internationaux, l'évolution de la réglementation numérique, la gouvernance des données et l'impact des technologies telles que l'intelligence artificielle, la *blockchain* et la cybersécurité. Tout au long du programme, l'analyse géopolitique et technologique est articulée autour d'une approche appliquée au développement et à la gestion de Software, préparant les professionnels à intervenir dans des systèmes complexes et des environnements numériques interconnectés.



66

*Vous analyserez les techniques les plus avancées pour garantir la sécurité et la fiabilité du Software”*

**Module 1.** Qualité du Software. Niveaux de développement TRL

- 1.1. Éléments influençant la qualité du Software (I). La dette technique
  - 1.1.1. La dette technique. Causes et conséquences
  - 1.1.2. Qualité du Software. Principes généraux
  - 1.1.3. Software sans principes et avec principes de qualité
    - 1.1.3.1. Conséquences
    - 1.1.3.2. La nécessité de l'application des principes de qualité dans le Software
  - 1.1.4. Qualité du Software. Typologie
  - 1.1.5. Software de qualité. Caractéristiques spécifiques
- 1.2. Éléments influençant la qualité du Software (II). Coûts associés
  - 1.2.1. Qualité du Software. Éléments d'influence
  - 1.2.2. Qualité du Software. Idées fausses
  - 1.2.3. Qualité du Software. Coûts associés
- 1.3. Modèles de qualité du Software (I). Gestion des connaissances
  - 1.3.1. Modèles de qualité générale
    - 1.3.1.1. Gestion de la qualité totale
    - 1.3.1.2. Modèle Européen d'Excellence Commerciale (EFQM)
    - 1.3.1.3. Modèle Six-sigma
  - 1.3.2. Modèles de Gestion des Connaissances
    - 1.3.2.1. Modèle Dyba
    - 1.3.2.2. Modèle SEKS
  - 1.3.3. Expérience du paradigme Factory et QIP
  - 1.3.4. Modèles de qualité d'usage (25010)
- 1.4. Modèles de qualité du Software (III). Qualité des données, des processus et des modèles SEI
  - 1.4.1. Modèle de qualité des données
  - 1.4.2. Modélisation du processus Software
  - 1.4.3. *Software & Systems Process Engineering Metamodel Specification (SPEM)*
  - 1.4.4. Modèles de la DIE
    - 1.4.4.1. CMMI
    - 1.4.4.2. SCAMPI
    - 1.4.4.3. IDEAL
- 1.5. Normes ISO de qualité du Software (I). Analyse des normes
  - 1.5.1. Normes ISO 9000
    - 1.5.1.1. Normes ISO 9000
    - 1.5.1.2. Famille de normes de qualité ISO (9000)
  - 1.5.2. Autres normes ISO relatives à la qualité
  - 1.5.3. Normes de modélisation de la qualité (ISO 2501)
  - 1.5.4. Normes de mesure de la qualité (ISO 2502n)
- 1.6. Normes ISO de qualité du Software (II). Exigences et évaluation
  - 1.6.1. Normes d'exigences de qualité (2503n)
  - 1.6.2. Normes sur l'évaluation de la qualité (2504n)
  - 1.6.3. ISO/IEC 24744:2007
- 1.7. Niveaux de développement TRL (I). Niveaux 1 à 4
  - 1.7.1. Niveau TRL
  - 1.7.2. Niveau 1 : principes de base
  - 1.7.3. Niveau 2 : concept et/ou application
  - 1.7.4. Niveau 3 : fonction analytique critique
  - 1.7.5. Niveau 4 : validation des composants dans un environnement de laboratoire
- 1.8. Niveaux de développement TRL (II). Niveaux de 5 à 9
  - 1.8.1. Niveau 5 : validation du composant dans un environnement pertinent
  - 1.8.2. Niveau 6 : modèle de système/sous-système
  - 1.8.3. Niveau 7 : démonstration en environnement réel
  - 1.8.4. Niveau 8 : système complet et certifié
  - 1.8.5. Niveau 9 : succès dans un environnement réel
- 1.9. Niveaux de développement TRL. Utilisations
  - 1.9.1. Exemple d'une entreprise avec un environnement de laboratoire
  - 1.9.2. Exemple d'une entreprise de R&D&I
  - 1.9.3. Exemple d'une entreprise de R&D&I industriel
  - 1.9.4. Exemple d'une entreprise commune laboratoire-ingénierie

- 1.10. Qualité du Software. Principaux détails
  - 1.10.1. Détails méthodologiques
  - 1.10.2. Détails techniques
  - 1.10.3. Détails sur la gestion des projets Software
    - 1.10.3.1. Qualité Systèmes d'information
    - 1.10.3.2. Qualité des produits Software
    - 1.10.3.3. Qualité du processus Software

## Module 2. Développement de projets Software. Documentation fonctionnelle et technique

- 2.1. Gestion de projets
  - 2.1.1. Gestion de projet en matière de qualité du Software
  - 2.1.2. Gestion de projets. Avantages
  - 2.1.3. Gestion de projets. Typologie
- 2.2. Méthodologie de la gestion de projet
  - 2.2.1. Méthodologie de la gestion de Projets
  - 2.2.2. Méthodologie de la gestion de projet Typologie
  - 2.2.3. Méthodologie dans la gestion de projets. Application
- 2.3. Phase d'identification des besoins
  - 2.3.1. Identification des besoins du projet
  - 2.3.2. Gestion des réunions de projet
  - 2.3.3. Documentation à fournir
- 2.4. Modèle
  - 2.4.1. Phase initiale
  - 2.4.2. Phase d'analyse
  - 2.4.3. Phase de construction
  - 2.4.4. Phase de test
  - 2.4.5. Livraison
- 2.5. Modèle de données à utiliser
  - 2.5.1. Détermination du nouveau modèle de données
  - 2.5.2. Identification du plan de migration des données
  - 2.5.3. Ensemble de données

- 2.6. Impact sur d'autres projets
    - 2.6.1. Impact d'un projet. Exemples
    - 2.6.2. Risques liés au projet
    - 2.6.3. Gestion des risques
  - 2.7. "Must" du projet
    - 2.7.1. Must du projet
    - 2.7.2. Identification du Must du projet
    - 2.7.3. Identification des points de mise en œuvre pour la réalisation d'un projet
  - 2.8. L'équipe de construction du projet
    - 2.8.1. Rôles à jouer en fonction du projet
    - 2.8.2. Contact avec les RH pour le recrutement
    - 2.8.3. Livrables et calendrier du projet
  - 2.9. Aspects techniques d'un projet *Software*
    - 2.9.1. Architecte du projet. Aspects Techniques
    - 2.9.2. Responsables techniques
    - 2.9.3. Construction du projet *Software*
    - 2.9.4. Évaluation de la qualité du code, sonar
  - 2.10. Livrables du projet
    - 2.10.1. Analyse fonctionnelle
    - 2.10.2. Modèles de données
    - 2.10.3. Diagrammes d'état
    - 2.10.4. Documentation technique
- 3.3. Norme ISO/IEC 15504
    - 3.3.1. Catégories de processus
    - 3.3.2. Processus de développement. Exemple
    - 3.3.3. Fragment de profil
    - 3.3.4. Étapes
  - 3.4. CMMI (*Capability Maturity Model Integration*)
    - 3.4.1. CMMI. Intégration du modèle de maturité de la capacité
    - 3.4.2. Modèles et zones. Typologie
    - 3.4.3. Domaines de processus
    - 3.4.4. Niveaux de capacité
    - 3.4.5. Gestion des processus
    - 3.4.6. Gestion de projet
  - 3.5. Gestion des changements et des référentiels
    - 3.5.1. Gestion des changements dans le *Software*
      - 3.5.1.1. Élément de configuration. Intégration continue
      - 3.5.1.2. Lignes
      - 3.5.1.3. Organigrammes
      - 3.5.1.4. Branches
    - 3.5.2. Référentiel
      - 3.5.2.1. Contrôle de la version
      - 3.5.2.2. Équipe de travail et utilisation du référentiel
      - 3.5.2.3. Intégration continue dans le référentiel
  - 3.6. Team Foundation Server (TFS)
    - 3.6.1. Installation et configuration
    - 3.6.2. Création d'un projet d'équipement
    - 3.6.3. Ajouter du contenu au contrôle de la source
    - 3.6.4. *TFS on Cloud*
  - 3.7. *Testing*
    - 3.7.1. Motivation pour les tests
    - 3.7.2. Test de vérification
    - 3.7.3. Tests bêta
    - 3.7.4. Mise en œuvre et maintenance

### Module 3. *Testing de Software. Automatisation des tests*

- 3.1. Modèles de qualité du *Software*
  - 3.1.1. Qualité du produit
  - 3.1.2. Qualité du processus
  - 3.1.3. Qualité de l'utilisation
- 3.2. Qualité du processus
  - 3.2.1. Qualité du processus
  - 3.2.2. Modèles de maturité
  - 3.2.3. Norme ISO 15504
    - 3.2.3.1. Objectifs
    - 3.2.3.2. Contexte
    - 3.2.3.3. Étapes

- 3.8. Essais de charge
  - 3.8.1. *Load testing*
  - 3.8.2. Tests avec *LoadView*
  - 3.8.3. Tests avec *K6 Cloud*
  - 3.8.4. Tests avec *Loader*
- 3.9. Tests unitaires, de stress et d'endurance
  - 3.9.1. Raison d'être des tests unitaires
  - 3.9.2. Outils de *Unit Testing*
  - 3.9.3. Motivation des tests de résistance
  - 3.9.4. Test en utilisant le *StressTesting*
  - 3.9.5. Motivation pour les tests de résistance
  - 3.9.6. Test à l'aide de *LoadRunner*
- 3.10. Évolutivité. Conception de *Software* évolutif
  - 3.10.1. Scalabilité et architecture du *Software*
  - 3.10.2. Indépendance entre les couches
  - 3.10.3. Couplage entre les couches. Modèles architecturaux

**Module 4.** Méthodologies de gestion de projets de *Software*. Méthodologies *waterfall* par rapport aux méthodologies agiles

- 4.1. Méthodologie *Waterfall*
  - 4.1.1. Méthodologie *Waterfall*
  - 4.1.2. Méthodologie *Waterfall*. Influence sur la qualité du *Software*
  - 4.1.3. Méthodologie *Waterfall*. Exemples
- 4.2. Méthodologie *Agile*
  - 4.2.1. Méthodologie *Agile*
  - 4.2.2. Méthodologie *Agile* Influence sur la qualité du *Software*
  - 4.2.3. Méthodologie *Agile* Exemples
- 4.3. Méthodologie *Scrum*
  - 4.3.1. Méthodologie *Scrum*
  - 4.3.2. Manifeste de *Scrum*
  - 4.3.3. Mise en œuvre de *Scrum*
- 4.4. Panel *Kanban*
  - 4.4.1. Méthode *Kanban*
  - 4.4.2. Panel *Kanban*
  - 4.4.3. Panel *Kanban*. Exemples d'application
- 4.5. Gestion de projet en *Waterfall*
  - 4.5.1. Phases d'un projet
  - 4.5.2. Vision dans un projet *Waterfall*
  - 4.5.3. Livrables à considérer
- 4.6. Gestion de projet en *Scrum*
  - 4.6.1. Phases d'un projet *Scrum*
  - 4.6.2. Vision dans un projet *Scrum*
  - 4.6.3. Produits livrables à considérer
- 4.7. *Waterfall* vs. Comparaison de *Scrum*
  - 4.7.1. Approche par projet pilote
  - 4.7.2. Projet utilisant *Waterfall*. Exemple
  - 4.7.3. Projet utilisant *Scrum*. Exemple
- 4.8. Aperçu des clients
  - 4.8.1. Documents dans un *Waterfall*
  - 4.8.2. Documents dans un *Scrum*
  - 4.8.3. Comparaison
- 4.9. Structure *Kanban*
  - 4.9.1. Histoires d'utilisateurs
  - 4.9.2. *Backlog*
  - 4.9.3. Analyse *Kanban*
- 4.10. Projets hybrides
  - 4.10.1. Construction du projet
  - 4.10.2. Gestion de projet
  - 4.10.3. Produits livrables à considérer

**Module 5.** TDD (*test driven development*). Conception de Software piloté par les tests

- 5.1. TDD. *Test Driven Development*
  - 5.1.1. TDD. *Test Driven Development*
  - 5.1.2. TDD. Influence du TDD sur la qualité
  - 5.1.3. Conception et développement pilotés par les tests. Exemples
- 5.2. Cycle TDD
  - 5.2.1. Choix d'une exigence
  - 5.2.2. Test. Typologie
    - 5.2.2.1. Tests unitaires
    - 5.2.2.2. Test d'intégration
    - 5.2.2.3. Preuves *End To End*
  - 5.2.3. Vérification des tests. Défaillances
  - 5.2.4. Création de la mise en œuvre
  - 5.2.5. Exécution de tests automatisés
  - 5.2.6. Élimination des doubles emplois
  - 5.2.7. Mise à jour de la liste des exigences
  - 5.2.8. Répétition du cycle TDD
  - 5.2.9. Cycle TDD. Exemple théoriques et pratiques
- 5.3. Stratégies de mise en œuvre du TDD
  - 5.3.1. Mise en œuvre fictive
  - 5.3.2. Mise en œuvre triangulaire
  - 5.3.3. Mise en œuvre évidente
- 5.4. TDD. Utilisation. Avantages et inconvénients
  - 5.4.1. Avantages de l'utilisation
  - 5.4.2. Limites d'utilisation
  - 5.4.3. Équilibre de la qualité dans la mise en œuvre
- 5.5. TDD. Bonnes pratiques
  - 5.5.1. Règles TDD
  - 5.5.2. Règle 1 : Faites un test préalable qui échoue avant de coder en production
  - 5.5.3. Règle 2 : ne pas écrire plus d'un test unitaire
  - 5.5.4. Règle 3 : ne pas écrire plus de code que nécessaire
  - 5.5.5. Erreurs et anti-modèles à éviter dans le TDD
- 5.6. Simulation d'un projet réel pour utiliser TDD (I)
  - 5.6.1. Aperçu du projet (Entreprise A)
  - 5.6.2. Application du TDD
  - 5.6.3. Exercices proposés
  - 5.6.4. Exercices *Feedback*
- 5.7. Simulation d'un projet réel pour utiliser le TDD (II)
  - 5.7.1. Aperçu du projet (Entreprise B)
  - 5.7.2. Application du TDD
  - 5.7.3. Exercices Proposés
  - 5.7.4. Exercices *Feedback*
- 5.8. Simulation d'un projet réel pour utiliser le TDD (III)
  - 5.8.1. Aperçu du projet (Entreprise C)
  - 5.8.2. Application du TDD
  - 5.8.3. Exercices Proposés
  - 5.8.4. Exercices *Feedback*
- 5.9. Alternatives au TDD. *Test Driven Development*
  - 5.9.1. TCR (*Test Commit Revert*)
  - 5.9.2. BDD (*Behavior Driven Development*)
  - 5.9.3. ATDD (*Acceptance Test Driven Development*)
  - 5.9.4. TDD. Comparaison théorique
- 5.10. TDD TCR, BDD et ATDD. Comparaison pratique
  - 5.10.1. Définition du problème
  - 5.10.2. Résoudre avec TCR
  - 5.10.3. Résoudre avec BDD
  - 5.10.4. Résoudre avec ATDD

## Module 6. DevOps. Gestion de qualité du Software

- 6.1. DevOps. Gestion de qualité du Software
  - 6.1.1. DevOps
  - 6.1.2. DevOps et qualité du Software
  - 6.1.3. DevOps. Avantages de la culture DevOps
- 6.2. DevOps. Relation avec Agile
  - 6.2.1. Livraison accélérée
  - 6.2.2. Qualité
  - 6.2.3. Réduction des coûts
- 6.3. Mise en œuvre de DevOps
  - 6.3.1. Identification des problèmes
  - 6.3.2. Mise en œuvre dans une entreprise
  - 6.3.3. Paramètres de mise en œuvre
- 6.4. Cycle de livraison du Software
  - 6.4.1. Méthodes de conception
  - 6.4.2. Conventions
  - 6.4.3. Feuille de route
- 6.5. Développement d'un code sans bogues
  - 6.5.1. Un code facile à maintenir
  - 6.5.2. Modèles de développement
  - 6.5.3. *Testing* du code
  - 6.5.4. Développement de Software au niveau du code. Bonnes pratiques
- 6.6. Automatisation
  - 6.6.1. Automatisation. Types de tests
  - 6.6.2. Coût de l'automatisation et de la maintenance
  - 6.6.3. Automatisation. Atténuer les erreurs
- 6.7. Déploiements
  - 6.7.1. Évaluation des objectifs
  - 6.7.2. Conception d'un processus automatique et adapté
  - 6.7.3. Retour d'information et réactivité

- 6.8. Gestion des incidents
  - 6.8.1. Préparation aux incidents
  - 6.8.2. Analyse et résolution des incidents
  - 6.8.3. Éviter les erreurs futures
- 6.9. Automatisation des déploiements
  - 6.9.1. Préparation des déploiements automatisés
  - 6.9.2. Évaluation automatique de l'état des processus
  - 6.9.3. Métriques et capacité de retour en arrière
- 6.10. Bonnes pratiques. Évolution de DevOps
  - 6.10.1. Guide des meilleures pratiques DevOps
  - 6.10.2. DevOps. Méthodologie pour l'équipe
  - 6.10.3. Éviter les niches

## Module 7. DevOps et intégration continue. Solutions pratiques avancées en matière de développement de Software

- 7.1. Flux de livraison du Software
  - 7.1.1. Identification des acteurs et des artefacts
  - 7.1.2. Conception du flux de livraison du Software
  - 7.1.3. Flux de livraison du Software. exigences entre les étapes
- 7.2. Automatisation des processus
  - 7.2.1. Intégration continue
  - 7.2.2. Intégration continue
  - 7.2.3. Configuration des environnements et gestion des secrets
- 7.3. Pipelines déclaratifs
  - 7.3.1. Différences entre les pipelines traditionnels, de type code et déclaratifs
  - 7.3.2. Pipelines déclaratifs
  - 7.3.3. Pipelines déclaratifs dans Jenkins
  - 7.3.4. Comparaison des fournisseurs d'intégration continue
- 7.4. Des portails de qualité et un retour d'information riche
  - 7.4.1. Portes de qualité
  - 7.4.2. Des normes de qualité avec des portes de qualité. Maintenance
  - 7.4.3. Exigences commerciales sur les demandes d'intégration

- 7.5. Gestion des artefacts
  - 7.5.1. Artefacts et cycle de vie
  - 7.5.2. Systèmes de stockage et de gestion des artefacts
  - 7.5.3. La sécurité dans la gestion des artefacts
- 7.6. Intégration continue
  - 7.6.1. Déploiement continu sous forme de conteneurs
  - 7.6.2. Déploiement continu avec PaaS
  - 7.6.3. Déploiement continu d'applications mobiles
- 7.7. Amélioration de l'exécution du pipeline : analyse statique et *Git Hooks*
  - 7.7.1. Analyse statique
  - 7.7.2. Règles de style de code
  - 7.7.3. *Git Hooks* et tests unitaires
  - 7.7.4. L'impact des infrastructures
- 7.8. Vulnérabilités dans les conteneurs
  - 7.8.1. Vulnérabilités dans les conteneurs
  - 7.8.2. Balayage d'images
  - 7.8.3. Rapports et alertes périodiques

**Module 8.** Conception de bases de données (BD). Standardisation et performance.

## Qualité du Software

- 8.1. Conception de bases de données
  - 8.1.1. Bases de données. Typologie
  - 8.1.2. Bases de données utilisées actuellement
    - 8.1.2.1. Relationnel
    - 8.1.2.2. Clé-valeur
    - 8.1.2.3. Basé sur le réseau
  - 8.1.3. Qualité des données
- 8.2. Conception d'un modèle entité-relation (I)
  - 8.2.1. Modèle entité-relation. Qualité et documentation
  - 8.2.2. Entités
    - 8.2.2.1. Entité forte
    - 8.2.2.2. Entité faible
- 8.2.3. Attributs
- 8.2.4. Ensemble de relations
  - 8.2.4.1. 1 à 1
  - 8.2.4.2. 1 à plusieurs
  - 8.2.4.3. De plusieurs à un
  - 8.2.4.4. Beaucoup à beaucoup
- 8.2.5. Clés
  - 8.2.5.1. Clé primaire
  - 8.2.5.2. Clé étrangère
  - 8.2.5.3. Clé primaire de l'entité faible
- 8.2.6. Restrictions
- 8.2.7. Cardinalité
- 8.2.8. Héritage
- 8.2.9. Agrégation
- 8.3. Modèle entité-relation (II). Outils
  - 8.3.1. Modèle entité-relation. Outils
  - 8.3.2. Modèle entité-relation. Exemple pratique
  - 8.3.3. Modèle entité-relation réalisable
    - 8.3.3.1. Échantillon visuel
    - 8.3.3.2. Échantillon en représentation de tableau
- 8.4. Normalisation (I) des bases de données (DB). Considérations sur la qualité du Software
  - 8.4.1. Normalisation et qualité des DB
  - 8.4.2. Dépendances
    - 8.4.2.1. Dépendance fonctionnelle
    - 8.4.2.2. Propriétés de la dépendance fonctionnelle
    - 8.4.2.3. Propriétés inférées
  - 8.4.3. Clés
- 8.5. Normalisation (II) de la base de données (BD). Formes normales et règles de Codd
  - 8.5.1. Formes normales
    - 8.5.1.1. Première forme normale (1FN)
    - 8.5.1.2. Deuxième forme normale (2FN)
    - 8.5.1.3. Troisième forme normale (3FN)

- 8.5.1.4. Forme normale de Boyce-Codd (BCNF)
  - 8.5.1.5. Quatrième forme normale (4FN)
  - 8.5.1.6. Cinquième forme normale (5FN)
  - 8.5.2. Les règles de Codd
    - 8.5.2.1. Règle 1 : Information
    - 8.5.2.2. Règle 2 : accès garanti
    - 8.5.2.3. Règle 3 : Traitement systématique des valeurs nulles
    - 8.5.2.4. Règle 4 : description de la base de données
    - 8.5.2.5. Règle 5 : Sous-langage intégral
    - 8.5.2.6. Règle n° 6 : Voir la mise à jour
    - 8.5.2.7. Règle 7 : Insertion et mise à jour
    - 8.5.2.8. Règle 8 : indépendance physique
    - 8.5.2.9. Règle 9 : indépendance logique
    - 8.5.2.10. Règle 10 : indépendance de l'intégrité
      - 8.5.2.10.1. Règles d'intégrité
    - 8.5.2.11. Règle 11 : distribution
    - 8.5.2.12. Règle 12 : Non-subversion
  - 8.5.3. Exemple pratique
  - 8.6. Entrepôt de données / système OLAP
    - 8.6.1. Entrepôt de données
    - 8.6.2. Tableau des faits
    - 8.6.3. Tableau des dimensions
    - 8.6.4. Création du système OLAP. Outils
  - 8.7. Performances des bases de données (DB)
    - 8.7.1. Optimisation de l'index
    - 8.7.2. Optimisation des requêtes
    - 8.7.3. Partitionnement des tables
  - 8.8. Simulation du projet réel pour la conception du DB (I)
    - 8.8.1. Aperçu du projet (Entreprise A)
    - 8.8.2. Application de la conception de bases de données
    - 8.8.3. Exercices proposés
    - 8.8.4. Exercices proposés. *Feedback*
  - 8.9. Simulation d'un projet réel pour la conception de BD (II)
    - 8.9.1. Aperçu du projet (Entreprise B)
    - 8.9.2. Application de la conception de bases de données
    - 8.9.3. Exercices proposés
    - 8.9.4. Exercices proposés. *Feedback*
  - 8.10. Pertinence de l'optimisation des bases de données dans la Qualité du Software
    - 8.10.1. Optimisation de la conception
    - 8.10.2. Optimisation du code de requête
    - 8.10.3. Optimisation du code des procédures stockées
    - 8.10.4. Influence des *Triggers* sur la qualité du Software. Recommandations d'utilisation
- Module 9.** Conception d'architectures évolutives. L'architecture dans le cycle de vie du Software
- 9.1. Conception d'architectures évolutives(I)
    - 9.1.1. Architectures évolutives
    - 9.1.2. Principes d'une architecture évolutive
      - 9.1.2.1. Fiable
      - 9.1.2.2. Évolutif
      - 9.1.2.3. Maintenable
    - 9.1.3. Types d'extensibilité
      - 9.1.3.1. Vertical
      - 9.1.3.2. Horizontal
      - 9.1.3.3. Combinaison
  - 9.2. Architectures de DDD (*Domain-Driven Design*)
    - 9.2.1. Le Modèle DDD. Orientation du domaine
    - 9.2.2. Couches, répartition des responsabilités et modèles de conception
    - 9.2.3. Le découplage comme base de la qualité

- 9.3. Conception d'architectures évolutives (II). Avantages, limites et stratégies de conception
  - 9.3.1. Architecture évolutive. Bénéfices
  - 9.3.2. Architecture évolutive. Limites
  - 9.3.3. Stratégies pour le développement d'architectures évolutives (Tableau descriptif)
- 9.4. Cycle de qualité du Software (I). Étapes
  - 9.4.1. Cycle de vie du Software
    - 9.4.1.1. Phase de planification
    - 9.4.1.2. Phase d'analyse
    - 9.4.1.3. Phase de conception
    - 9.4.1.4. Phase de mise en œuvre
    - 9.4.1.5. Phase de test
    - 9.4.1.6. Phase d'installation/déploiement
    - 9.4.1.7. Phase d'utilisation et de maintenance
- 9.5. Modèles de cycle de vie du Software
  - 9.5.1. Modèle en cascade
  - 9.5.2. Modèle répétitif
  - 9.5.3. Modèle en spirale
  - 9.5.4. Modèle *Big Bang*
- 9.6. Cycle de vie du Software (II). Automatisation
  - 9.6.1. Cycles de vie du développement de Software. Solutions
    - 9.6.1.1. Intégration continue et développement continu (CI/CD)
    - 9.6.1.2. Méthodologies Agiles
    - 9.6.1.3. DevOps / opérations de production
  - 9.6.2. Tendances futures
  - 9.6.3. Exemples pratiques
- 9.7. Architecture du Software dans le cycle de vie du Software
  - 9.7.1. Bénéfices
  - 9.7.2. Limites
  - 9.7.3. Outils
- 9.8. Simulation d'un projet réel pour la conception d'une architecture Software (I)
  - 9.8.1. Aperçu du projet (Entreprise A)
  - 9.8.2. Application de la conception architecturale du Software
  - 9.8.3. Exercices Proposés
  - 9.8.4. Exercices Proposés. *Feedback*
- 9.9. Simulation d'un projet réel pour la conception de l'architecture du Software (II)
  - 9.9.1. Aperçu du projet (Entreprise B)
  - 9.9.2. Application de la conception architecturale du Software
  - 9.9.3. Exercices Proposés
  - 9.9.4. Exercices Proposés. *Feedback*
- 9.10. Simulation d'un projet réel pour la conception de l'architecture du Software (III)
  - 9.10.1. Aperçu du projet (Entreprise C)
  - 9.10.2. Application de la conception architecturale du Software
  - 9.10.3. Exercices Proposés
  - 9.10.4. Exercices Proposés. *Feedback*

**Module 10.** Critères de qualité ISO, IEC 9126. Mesures de la qualité du Software

- 10.1. Critères de qualité Norme ISO, IEC 9126
  - 10.1.1. Critères de qualité
  - 10.1.2. Qualité du Software. Justification. Norme ISO, IEC 9126
  - 10.1.3. La mesure de la qualité du Software comme indicateur clé
- 10.2. Critères de qualité du Software. Caractéristiques
  - 10.2.1. Fiabilité
  - 10.2.2. Fonctionnalité
  - 10.2.3. Efficacité
  - 10.2.4. Utilisabilité
  - 10.2.5. Maintenance
  - 10.2.6. Portabilité
  - 10.2.7. Sécurité

- 10.3. Norme ISO, CEI 9126 (I). Présentation
  - 10.3.1. Description de la Norme ISO, IEC 9126
  - 10.3.2. Fonctionnalité
  - 10.3.3. Fiabilité
  - 10.3.4. Utilisabilité
  - 10.3.5. Maintenance
  - 10.3.6. Portabilité
  - 10.3.7. Qualité de l'utilisation
  - 10.3.8. Mesures de la qualité du Software
  - 10.3.9. Les mesures de la qualité dans la norme ISO 9126
- 10.4. Norme ISO, CEI 9126 (II). Modèles de McCall et Boehm
  - 10.4.1. Modèle de McCall : facteurs de qualité
  - 10.4.2. Modèle Boehm
  - 10.4.3. Niveau intermédiaire. Caractéristiques
- 10.5. Mesure de la qualité du Software (I). Éléments
  - 10.5.1. Mesure
  - 10.5.2. Métriques
  - 10.5.3. Indicateur
    - 10.5.3.1. Types d'indicateurs
  - 10.5.4. Mesures et modèles
  - 10.5.5. Portée des mesures du Software
  - 10.5.6. Classification des indicateurs de Software
- 10.6. Mesure de la qualité du Software (II). Pratique de la mesure
  - 10.6.1. Collecte de données métriques
  - 10.6.2. Mesure des attributs internes du produit
  - 10.6.3. Mesure des attributs externes du produit
  - 10.6.4. Mesure des ressources
  - 10.6.5. Métriques pour les systèmes orientés objet
- 10.7. Conception d'un indicateur unique de qualité du Software
  - 10.7.1. Indicateur unique en tant que scoreur global
  - 10.7.2. Développement, justification et application des indicateurs
  - 10.7.3. Exemples d'application. Besoin de connaître le détail
- 10.8. Simulation d'un projet réel pour la mesure de la qualité (I)
  - 10.8.1. Aperçu du projet (Entreprise A)
  - 10.8.2. Application de la mesure de la qualité
  - 10.8.3. Exercices Proposés
  - 10.8.4. Exercices Proposés. *Feedback*
- 10.9. Simulation d'un projet réel pour la mesure de la qualité (II)
  - 10.9.1. Aperçu du projet (Entreprise B)
  - 10.9.2. Application de la mesure de la qualité
  - 10.9.3. Exercices Proposés
  - 10.9.4. Exercices Proposés. *Feedback*
- 10.10. Simulation d'un projet réel pour la mesure de la qualité (III)
  - 10.10.1. Aperçu du projet (Entreprise C)
  - 10.10.2. Application de la mesure de la qualité
  - 10.10.3. Exercices Proposés
  - 10.10.4. Exercices Proposés. *Feedback*

“

*Vous gérerez des projets Software dans une optique d'amélioration continue et de création de valeur durable”*

04

# Objectifs pédagogiques

Le programme a pour objectif principal de fournir une compréhension approfondie et intégrée de l'environnement mondial du point de vue du développement de Software et des technologies de l'information. À cette fin, il vise à doter l'étudiant des outils conceptuels et pratiques nécessaires pour analyser l'influence des cadres réglementaires, des crises technologiques et des organismes internationaux sur l'innovation numérique. En outre, l'objectif est d'encourager une vision critique des défis éthiques, de la sécurité informatique et de l'évolution de la gouvernance numérique. Ainsi, la capacité à concevoir, adapter et gérer des solutions technologiques dans des scénarios de plus en plus complexes est encouragée.



66

*Il applique des méthodologies agiles et des outils DevOps pour optimiser les processus et accélérer les cycles de développement dans des environnements réels”*



## Objectifs généraux

- Développez les critères, les tâches et les méthodologies avancées pour comprendre la pertinence d'un Travail axé sur la Qualité
- Analyser les facteurs clés de la Qualité d'un Projet Software
- Développer les aspects réglementaires pertinents
- Mise en œuvre de processus DevOps et de Systèmes pour l'Assurance Qualité
- Réduire la Dette Technique des Projets avec une approche de Qualité plutôt qu'une approche basée sur l'économie et les délais courts
- Fournir à l'étudiant le savoir-faire pour être capable de Mesurer et de Quantifier la Qualité d'un Projet Software

“

*Vous intégrerez des connaissances en cybersécurité, en tests et en contrôle qualité afin de garantir la solidité et la fiabilité de vos développements”*





## Objectifs spécifiques

### Module 1. Qualité du Software. Niveaux de développement TRL

- ♦ Développer de manière claire et concise les éléments qui englobent la qualité du Software
- ♦ Appliquer les modèles et les normes en fonction du système, du produit et du processus Software
- ♦ Approfondir les normes de qualité ISO appliquées tant en général que dans des parties spécifiques
- ♦ Appliquer les normes en fonction de la portée de l'environnement (local, national, international)
- ♦ Examiner les niveaux de maturité TRL et les adapter aux différentes parties du projet Software à traiter
- ♦ Acquérir une capacité d'abstraction pour appliquer un ou plusieurs critères d'éléments et niveaux de qualité du Software

### Module 2. Développement de projets Software. Documentation fonctionnelle et technique

- ♦ Déterminer l'influence de la Gestion de Projet sur la Qualité
- ♦ Développer les différentes Phases d'un Projet
- ♦ Différencier les Concepts de Qualité inhérents à la Documentation Fonctionnelle et Technique
- ♦ Analyser la Phase de collecte des Besoins, la phase d'Analyse, la gestion de l'Équipe et la Phase de Construction
- ♦ Établir les différentes Méthodologies de Gestion des Projets Software
- ♦ Générer des critères pour décider quelle est la Méthodologie la plus appropriée en fonction du type de projet

**Module 3. Testing de Software. Automatisation des tests**

- ◆ Établir les différences entre la qualité du produit, la qualité du processus et la qualité d'utilisation
- ◆ Comprendre la norme ISO/IEC 15504
- ◆ Déterminer les détails du CMMI
- ◆ Pour connaître les clés de l'intégration continue, les référentiels et leurs répercussions sur une équipe de développement de *Software*
- ◆ Établir la pertinence de l'intégration de référentiels pour les projets de *Software*. Apprenez à les créer avec TFS
- ◆ Assimiler l'importance de l'évolutivité du *Software* dans la conception et le développement des systèmes d'information

**Module 4. Méthodologies de gestion de projets de Software. Méthodologies waterfall par rapport aux méthodologies agiles**

- ◆ Déterminer en quoi consiste la méthodologie *Waterfall*
- ◆ Approfondir la Méthodologie *SCRUM*
- ◆ Établir les Différences entre *Waterfall* et *SCRUM*
- ◆ Pour préciser les différences entre les Méthodologies *Waterfall* et *SCRUM* et comment le client le voit
- ◆ Examen du Panel Kanban
- ◆ Mise en place d'un même projet avec *Waterfall* et *SCRUM*

**Module 5. TDD (*test driven development*). Conception de *Software* piloté par les tests**

- ◆ Connaître l'application pratique du TDD et ses possibilités pour la réalisation future de tests d'un projet *Software*
- ◆ Compléter les cas de simulation réels proposés, comme un apprentissage continu de ce concept TDD
- ◆ Analyser, dans les cas de simulation, dans quelle mesure les tests peuvent réussir ou échouer d'un point de vue constructif
- ◆ Déterminer les alternatives au TDD, en effectuant une analyse comparative entre elles

**Module 6. DevOps. Gestion de qualité du *Software***

- ◆ Analyser les défauts d'un processus traditionnel
- ◆ Évaluer les solutions possibles et choisir la plus appropriée
- ◆ Comprendre les besoins de l'entreprise et leur impact sur la mise en œuvre
- ◆ Évaluer les coûts des améliorations à mettre en œuvre
- ◆ Développer un Cycle de Vie du *Software* évolutif, adapté aux besoins réels
- ◆ Anticipez les erreurs possibles et évitez-les dès le processus de conception

**Module 7. DevOps et intégration continue. Solutions pratiques avancées en matière de développement de Software**

- Identifier les étapes du cycle de développement et de livraison du Software adaptées aux cas particuliers
- Concevoir un processus de livraison de Software utilisant l'intégration continue
- Construire et mettre en œuvre l'intégration et le déploiement continus sur la base de sa conception précédente
- Établir des points de contrôle de qualité automatiques sur chaque livraison de Software
- Maintenir un processus de livraison de Software automatisé et robuste
- Adapter les besoins futurs au processus d'intégration et de déploiement continus

**Module 8. Conception de bases de données (BD). Standardisation et performance. Qualité du Software**

- Évaluer l'utilisation du modèle entité-relation pour la conception préliminaire d'une base de données
- Appliquez une entité, un attribut, une clé, etc. afin pour une meilleure intégrité des données
- Évaluer les dépendances, les formes et les règles de la normalisation des bases de données
- Se spécialiser dans l'exploitation d'un système d'entrepôt de données OLAP, en développant et en utilisant des tables de faits et de dimensions
- Déterminer les points clés pour les performances de la base de données
- Réaliser des cas de simulation réels proposés comme expérience d'apprentissage continu en matière de conception, de normalisation et de performance des bases de données

**Module 9. Conception d'architectures évolutives. L'architecture dans le cycle de vie du Software**

- Développer le concept d'Architecture du Software et ses caractéristiques
- Déterminer les différents types d'évolutivité dans l'Architecture du Software
- Analyser les différents niveaux qui peuvent intervenir dans l'évolutivité du Web
- Acquérir des connaissances spécialisées sur le concept du Cycle de Vie du Software, ses étapes et ses modèles
- Déterminer l'impact d'une architecture sur le Cycle de Vie du Software, avec ses avantages, ses limites et les outils de soutien
- Réaliser les cas de simulation réels proposés, en tant qu'apprentissage continu de l'Architecture et du Cycle de Vie du Software

**Module 10. Critères de qualité ISO, IEC 9126. Mesures de la qualité du Software**

- Développer le concept de critères de qualité et les aspects pertinents
- Examiner la norme ISO/IEC 9126, ses principaux aspects et ses indicateurs
- Analyser les différentes métriques d'un projet Software pour répondre aux évaluations convenues
- Examiner les attributs internes et externes à prendre en compte dans la qualité d'un projet Software
- Distinguer les métriques en fonction du type de programmation (structurée, orientée objet, en couches...)
- Réalisation de cas réels de simulation comme processus d'apprentissage continu en matière de mesure de la qualité

05

# Opportunités de carrière

Ce programme forme des profils prêts à faire une entrée remarquée dans le secteur du Software et des technologies mondiales. Au-delà des rôles traditionnels, il ouvre la voie à des postes émergents tels que stratège en innovation numérique, analyste d'environnements technologiques complexes ou coordinateur de projets transversaux dans des environnements internationaux. De même, la maîtrise des cadres réglementaires et des outils numériques ouvre la porte à des collaborations avec des institutions ou des *start-ups* technologiques. Ainsi, les diplômés ne se contentent pas d'intégrer le marché : ils le redéfinissent en apportant une vision critique, globale et actualisée du développement technologique.



66

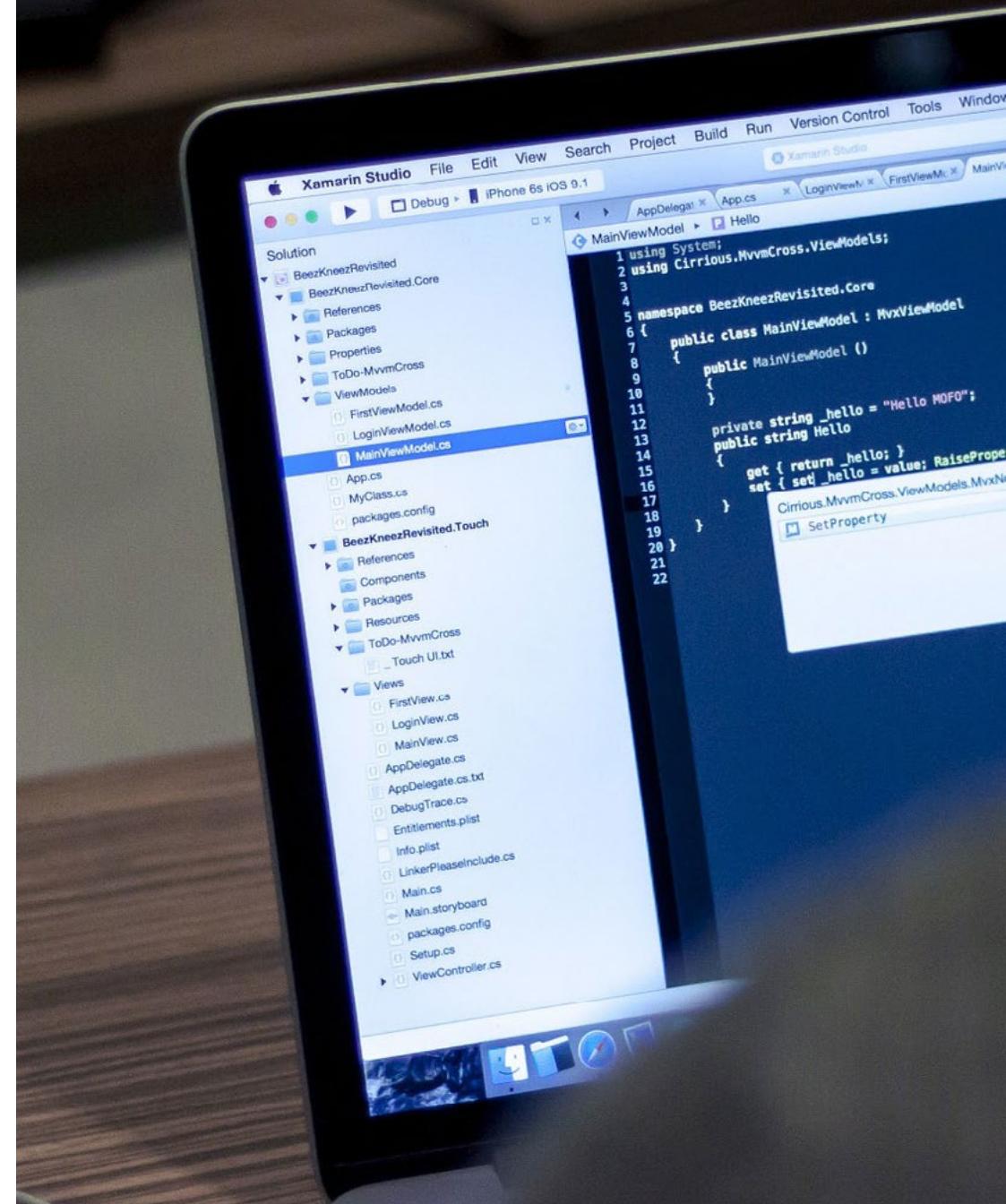
Spécialisez-vous dans les architectures de microservices et de conteneurs pour mener à bien des projets évolutifs et efficaces dans le cloud”

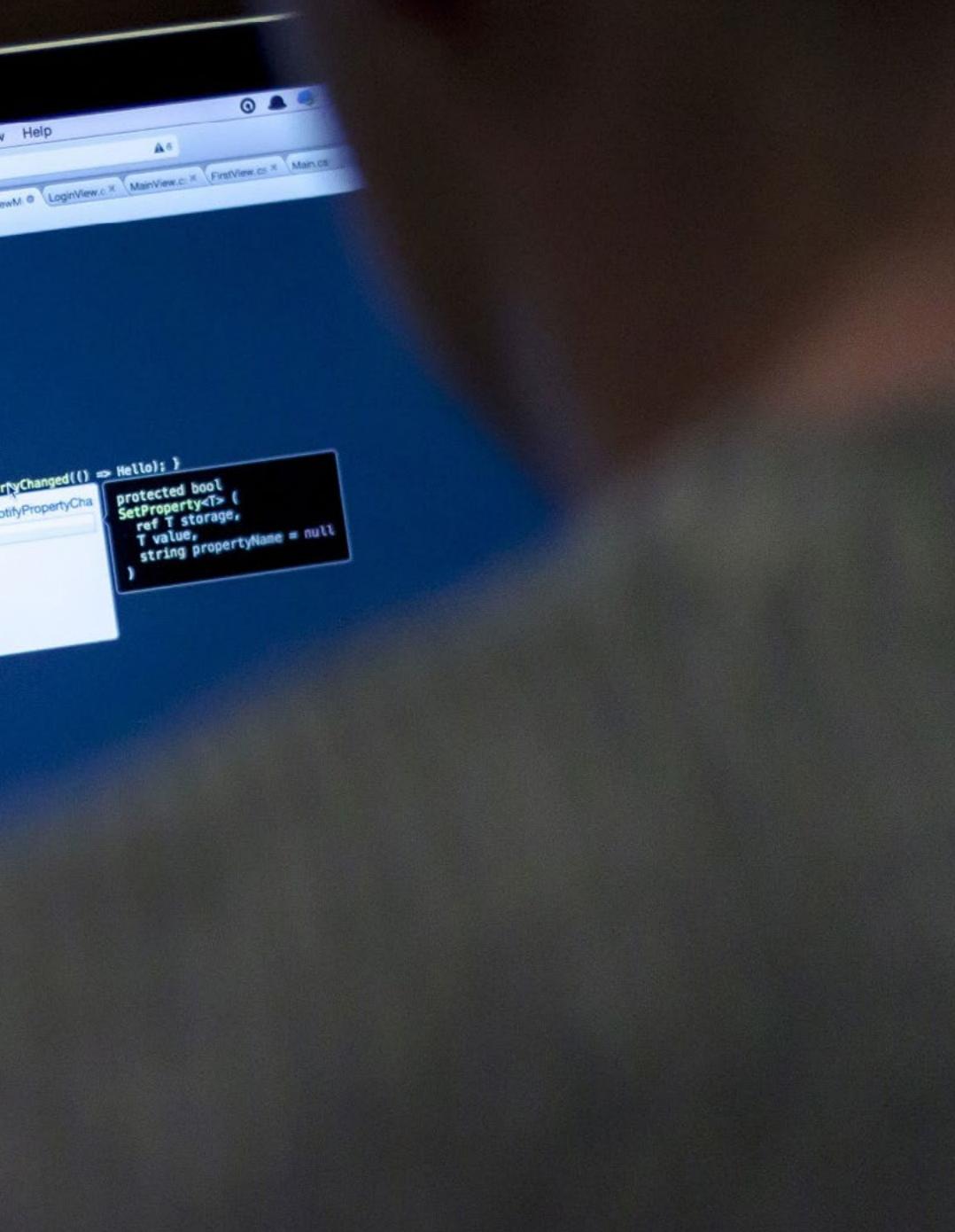
### Profil des diplômés

À l'issue de ce programme, le diplômé se distingue par sa capacité à intégrer des connaissances techniques, une pensée analytique et une vision stratégique dans le domaine du Software. Il maîtrise les langages, les outils et les environnements de développement, mais sait également interpréter les cadres réglementaires, diriger des équipes multidisciplinaires et prendre des décisions dans des contextes de changement technologique accéléré. Grâce à une formation qui combine théorie de pointe et application pratique, il acquiert les compétences nécessaires pour concevoir des solutions innovantes, optimiser les processus numériques et s'adapter avec agilité aux défis de l'écosystème technologique mondial. Il ne se contente pas de résoudre des problèmes : il anticipe les tendances et génère un impact.

*Vous souhaitez exercer le métier d'Ingénieur en Automatisation des Tests ? Réalisez votre rêve en quelques mois seulement grâce à ce diplôme universitaire.*

- ◆ **Pensée critique et analytique** : Capacité à évaluer des informations complexes, à identifier des modèles et à prendre des décisions éclairées dans des environnements technologiques dynamiques
- ◆ **Résolution de problèmes** : Capacité à aborder les défis sous plusieurs angles, en proposant des solutions innovantes et durables
- ◆ **Travail collaboratif** : Capacité à s'intégrer dans des équipes multidisciplinaires, à communiquer efficacement et à contribuer à la réalisation d'objectifs communs
- ◆ **Apprentissage autonome et continu** : Disposition à mettre à jour ses connaissances et à s'adapter à de nouveaux outils, langages et méthodologies tout au long de sa carrière professionnelle





À l'issue de ce programme, vous serez en mesure d'utiliser vos connaissances et vos compétences dans les postes suivants :

- 1. Ingénieur de Software**: Conçoit, développe et assure la maintenance d'applications et de systèmes informatiques efficaces et évolutifs.
- 2. Architecte Logiciel** : Définit la structure technique de systèmes complexes, en garantissant leur performance, leur sécurité et leur maintenabilité.
- 3. Développeur Backend** : Crée la logique de fonctionnement des applications, en gérant les bases de données, les serveurs et les API.
- 4. Développeur Frontend** : Conçoit et met en œuvre des interfaces interactives, garantissant une expérience utilisateur intuitive et réactive.
- 5. Spécialiste en Intégration de Systèmes** : Connecte et synchronise diverses plateformes et services afin de créer des écosystèmes numériques interopérables.
- 6. DevOps Engineer** : Automatise les processus de développement, de test et de déploiement afin d'améliorer l'efficacité du cycle de vie du Software.
- 7. Consultant en Solutions Technologiques** : Conseille les entreprises dans le choix, la mise en œuvre et l'amélioration de systèmes informatiques personnalisés.
- 8. Analyste Système** : Examine les exigences technologiques, modélise des solutions et s'assure que le Software répond aux besoins de l'utilisateur.

“

*Accédez à un large éventail d'opportunités professionnelles en tant qu'ingénieur du Software, architecte système, consultant technique ou responsable du développement”*

06

## Licences de logiciels incluses

TECH est une référence dans le monde universitaire pour associer les dernières technologies aux méthodologies d'enseignement afin d'améliorer le processus d'enseignement-apprentissage. À cette fin, elle a établi un réseau d'alliances qui lui permet d'avoir accès aux outils logiciels les plus avancés du monde professionnel.



66

*Lorsque vous vous inscrirez, vous recevrez, tout à fait gratuitement, les références pour l'utilisation académique des applications logicielles professionnelles suivantes"*

TECH a établi un réseau d'alliances professionnelles avec les principaux fournisseurs de logiciels appliqués à différents domaines professionnels. Ces alliances permettent à TECH d'avoir accès à l'utilisation de centaines d'applications informatiques et de licences de software afin de les rapprocher de ses étudiants.

Les licences de logiciels pour un universitaire permettront aux étudiants d'utiliser les applications informatiques les plus avancées dans leur domaine professionnel, afin qu'ils puissent les connaître et apprendre à les maîtriser sans avoir à engager de frais. TECH se chargera de la procédure contractuelle afin que les étudiants puissent les utiliser de manière illimitée pendant la durée de leurs études dans le cadre du programme de Mastère Spécialisé en Qualité du Software, et ce de manière totalement gratuite.

TECH vous donnera un accès gratuit à l'utilisation des applications logicielles suivantes :



#### Google Career Launchpad

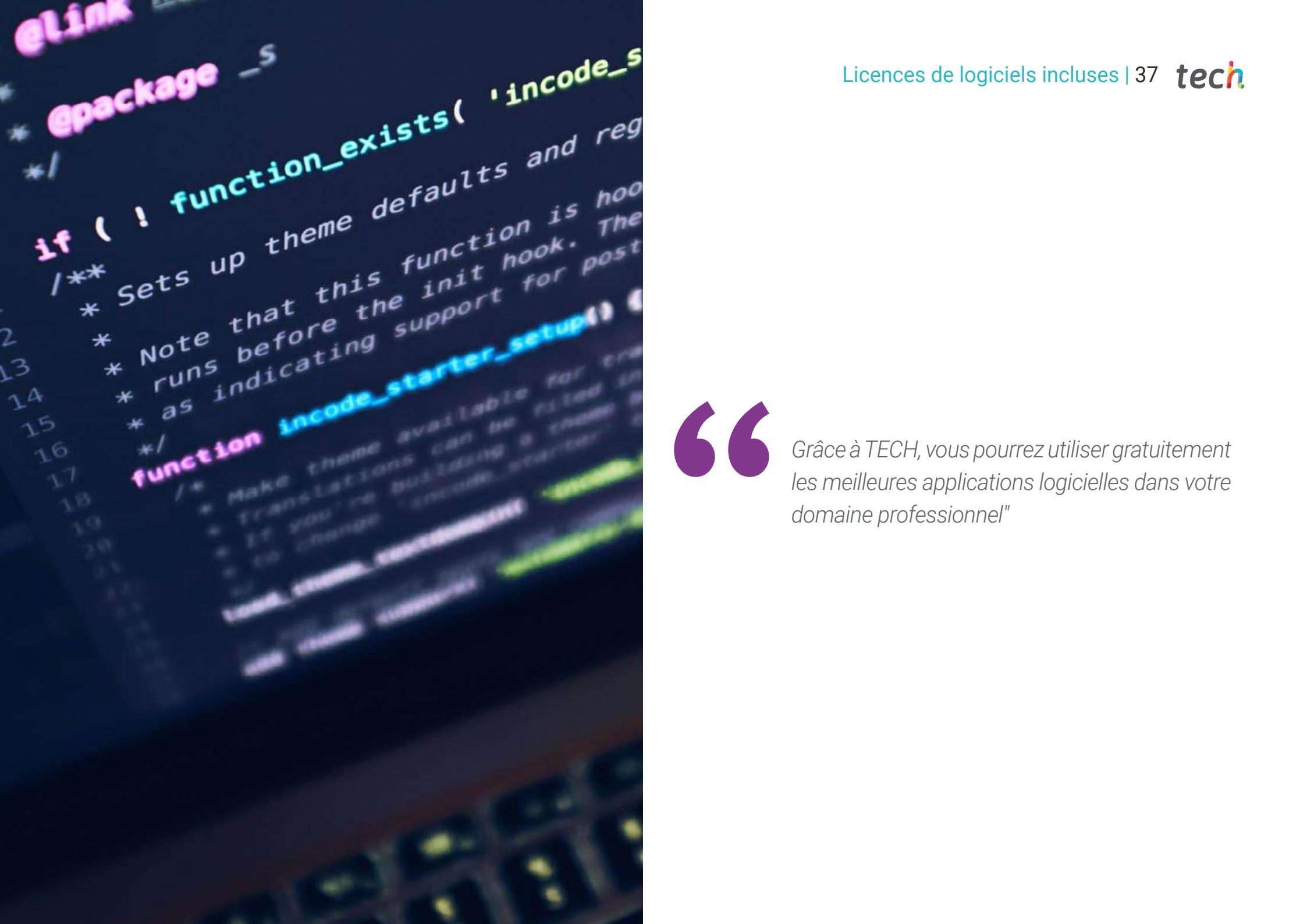
**Google Career Launchpad** est une solution pour développer des compétences numériques en technologie et en analyse de données. D'une valeur estimée à **5 000 dollars**, il est inclus gratuitement dans le programme universitaire de TECH, donnant accès à des laboratoires interactifs et à des certifications reconnues par l'industrie.

Cette plateforme combine la formation technique avec des études de cas, en utilisant des technologies telles que BigQuery et Google AI. Elle offre des environnements simulés pour expérimenter avec des données réelles, ainsi qu'un réseau d'experts pour un accompagnement personnalisé.

#### Fonctionnalités remarquables :

- ◆ **Cours spécialisés** : contenu actualisé sur le cloud computing, le machine learning et l'analyse de données
- ◆ **Laboratoires en direct** : pratique avec de vrais outils Google Cloud sans configuration supplémentaire
- ◆ **Certifications intégrées** : préparation aux examens officiels avec validité internationale
- ◆ **Mentorat professionnel** : sessions avec des experts Google et des partenaires technologiques
- ◆ **Projets collaboratifs** : défis basés sur des problèmes réels d'entreprises de premier plan

En conclusion, **Google Career Launchpad** connecte les utilisateurs aux dernières technologies du marché, facilitant leur insertion dans des domaines tels que l'intelligence artificielle et la science des données avec des titres de compétences soutenus par l'industrie.



“

Grâce à TECH, vous pourrez utiliser gratuitement les meilleures applications logicielles dans votre domaine professionnel”

07

# Méthodologie d'étude

TECH est la première université au monde à combiner la méthodologie des **case studies** avec **Relearning**, un système d'apprentissage 100% en ligne basé sur la répétition guidée.

Cette stratégie d'enseignement innovante est conçue pour offrir aux professionnels la possibilité d'actualiser leurs connaissances et de développer leurs compétences de manière intensive et rigoureuse. Un modèle d'apprentissage qui place l'étudiant au centre du processus académique et lui donne le rôle principal, en s'adaptant à ses besoins et en laissant de côté les méthodologies plus conventionnelles.



“

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

## L'étudiant: la priorité de tous les programmes de TECH

Dans la méthodologie d'étude de TECH, l'étudiant est le protagoniste absolu. Les outils pédagogiques de chaque programme ont été sélectionnés en tenant compte des exigences de temps, de disponibilité et de rigueur académique que demandent les étudiants d'aujourd'hui et les emplois les plus compétitifs du marché.

Avec le modèle éducatif asynchrone de TECH, c'est l'étudiant qui choisit le temps qu'il consacre à l'étude, la manière dont il décide d'établir ses routines et tout cela dans le confort de l'appareil électronique de son choix. L'étudiant n'a pas besoin d'assister à des cours en direct, auxquels il ne peut souvent pas assister. Les activités d'apprentissage se dérouleront à votre convenance. Vous pouvez toujours décider quand et où étudier.

“

*À TECH, vous n'aurez PAS de cours en direct  
(auxquelles vous ne pourrez jamais assister)"*





## Les programmes d'études les plus complets au niveau international

TECH se caractérise par l'offre des itinéraires académiques les plus complets dans l'environnement universitaire. Cette exhaustivité est obtenue grâce à la création de programmes d'études qui couvrent non seulement les connaissances essentielles, mais aussi les dernières innovations dans chaque domaine.

Grâce à une mise à jour constante, ces programmes permettent aux étudiants de suivre les évolutions du marché et d'acquérir les compétences les plus appréciées par les employeurs. Ainsi, les diplômés de TECH reçoivent une préparation complète qui leur donne un avantage concurrentiel significatif pour progresser dans leur carrière.

De plus, ils peuvent le faire à partir de n'importe quel appareil, PC, tablette ou smartphone.

“

*Le modèle de TECH est asynchrone, de sorte que vous pouvez étudier sur votre PC, votre tablette ou votre smartphone où vous voulez, quand vous voulez et aussi longtemps que vous le voulez”*

## Case studies ou Méthode des cas

La méthode des cas est le système d'apprentissage le plus utilisé par les meilleures écoles de commerce du monde. Développée en 1912 pour que les étudiants en Droit n'apprennent pas seulement le droit sur la base d'un contenu théorique, sa fonction était également de leur présenter des situations réelles et complexes. De cette manière, ils pouvaient prendre des décisions en connaissance de cause et porter des jugements de valeur sur la manière de les résoudre. Elle a été établie comme méthode d'enseignement standard à Harvard en 1924.

Avec ce modèle d'enseignement, ce sont les étudiants eux-mêmes qui construisent leurs compétences professionnelles grâce à des stratégies telles que *Learning by doing* ou le *Design Thinking*, utilisées par d'autres institutions renommées telles que Yale ou Stanford.

Cette méthode orientée vers l'action sera appliquée tout au long du parcours académique de l'étudiant avec TECH. Vous serez ainsi confronté à de multiples situations de la vie réelle et devrez intégrer des connaissances, faire des recherches, argumenter et défendre vos idées et vos décisions. Il s'agissait de répondre à la question de savoir comment ils agiraient lorsqu'ils seraient confrontés à des événements spécifiques complexes dans le cadre de leur travail quotidien.



## Méthode *Relearning*

Chez TECH, les case studies sont complétées par la meilleure méthode d'enseignement 100% en ligne: le *Relearning*.

Cette méthode s'écarte des techniques d'enseignement traditionnelles pour placer l'apprenant au centre de l'équation, en lui fournissant le meilleur contenu sous différents formats. De cette façon, il est en mesure de revoir et de répéter les concepts clés de chaque matière et d'apprendre à les appliquer dans un environnement réel.

Dans le même ordre d'idées, et selon de multiples recherches scientifiques, la répétition est le meilleur moyen d'apprendre. C'est pourquoi TECH propose entre 8 et 16 répétitions de chaque concept clé au sein d'une même leçon, présentées d'une manière différente, afin de garantir que les connaissances sont pleinement intégrées au cours du processus d'étude.

*Le Relearning vous permettra d'apprendre plus facilement et de manière plus productive tout en développant un esprit critique, en défendant des arguments et en contrastant des opinions: une équation directe vers le succès.*



## Un Campus Virtuel 100% en ligne avec les meilleures ressources didactiques

Pour appliquer efficacement sa méthodologie, TECH se concentre à fournir aux diplômés du matériel pédagogique sous différents formats: textes, vidéos interactives, illustrations et cartes de connaissances, entre autres. Tous ces supports sont conçus par des enseignants qualifiés qui axent leur travail sur la combinaison de cas réels avec la résolution de situations complexes par la simulation, l'étude de contextes appliqués à chaque carrière professionnelle et l'apprentissage basé sur la répétition, par le biais d'audios, de présentations, d'animations, d'images, etc.

Les dernières données scientifiques dans le domaine des Neurosciences soulignent l'importance de prendre en compte le lieu et le contexte d'accès au contenu avant d'entamer un nouveau processus d'apprentissage. La possibilité d'ajuster ces variables de manière personnalisée aide les gens à se souvenir et à stocker les connaissances dans l'hippocampe pour une rétention à long terme. Il s'agit d'un modèle intitulé *Neurocognitive context-dependent e-learning* qui est sciemment appliqué dans le cadre de ce diplôme universitaire.

D'autre part, toujours dans le but de favoriser au maximum les contacts entre mentors et mentorés, un large éventail de possibilités de communication est offert, en temps réel et en différé (messagerie interne, forums de discussion, service téléphonique, contact par courrier électronique avec le secrétariat technique, chat et vidéoconférence).

De même, ce Campus Virtuel très complet permettra aux étudiants TECH d'organiser leurs horaires d'études en fonction de leurs disponibilités personnelles ou de leurs obligations professionnelles. De cette manière, ils auront un contrôle global des contenus académiques et de leurs outils didactiques, mis en fonction de leur mise à jour professionnelle accélérée.

“

*Le mode d'étude en ligne de ce programme vous permettra d'organiser votre temps et votre rythme d'apprentissage, en l'adaptant à votre emploi du temps"*

### **L'efficacité de la méthode est justifiée par quatre acquis fondamentaux:**

1. Les étudiants qui suivent cette méthode parviennent non seulement à assimiler les concepts, mais aussi à développer leur capacité mentale au moyen d'exercices pour évaluer des situations réelles et appliquer leurs connaissances.
2. L'apprentissage est solidement traduit en compétences pratiques ce qui permet à l'étudiant de mieux s'intégrer dans le monde réel.
3. L'assimilation des idées et des concepts est rendue plus facile et plus efficace, grâce à l'utilisation de situations issues de la réalité.
4. Le sentiment d'efficacité de l'effort investi devient un stimulus très important pour les étudiants, qui se traduit par un plus grand intérêt pour l'apprentissage et une augmentation du temps passé à travailler sur le cours.



## La méthodologie universitaire la mieux évaluée par ses étudiants

Les résultats de ce modèle académique innovant sont visibles dans les niveaux de satisfaction générale des diplômés de TECH.

L'évaluation par les étudiants de la qualité de l'enseignement, de la qualité du matériel, de la structure du cours et des objectifs est excellente. Il n'est pas surprenant que l'institution soit devenue l'université la mieux évaluée par ses étudiants selon l'indice global score, obtenant une note de 4,9 sur 5.

*Accédez aux contenus de l'étude depuis n'importe quel appareil disposant d'une connexion Internet (ordinateur, tablette, smartphone) grâce au fait que TECH est à la pointe de la technologie et de l'enseignement.*

*Vous pourrez apprendre grâce aux avantages offerts par les environnements d'apprentissage simulés et à l'approche de l'apprentissage par observation: le Learning from an expert.*

Ainsi, le meilleur matériel pédagogique, minutieusement préparé, sera disponible dans le cadre de ce programme:



#### Matériel didactique

Tous les contenus didactiques sont créés par les spécialistes qui enseignent les cours. Ils ont été conçus en exclusivité pour le programme afin que le développement didactique soit vraiment spécifique et concret.

Ces contenus sont ensuite appliqués au format audiovisuel afin de mettre en place notre mode de travail en ligne, avec les dernières techniques qui nous permettent de vous offrir une grande qualité dans chacune des pièces que nous mettrons à votre service.



#### Pratique des aptitudes et des compétences

Vous effectuerez des activités visant à développer des compétences et des aptitudes spécifiques dans chaque domaine. Pratiques et dynamiques permettant d'acquérir et de développer les compétences et les capacités qu'un spécialiste doit acquérir dans le cadre de la mondialisation dans laquelle nous vivons.



#### Résumés interactifs

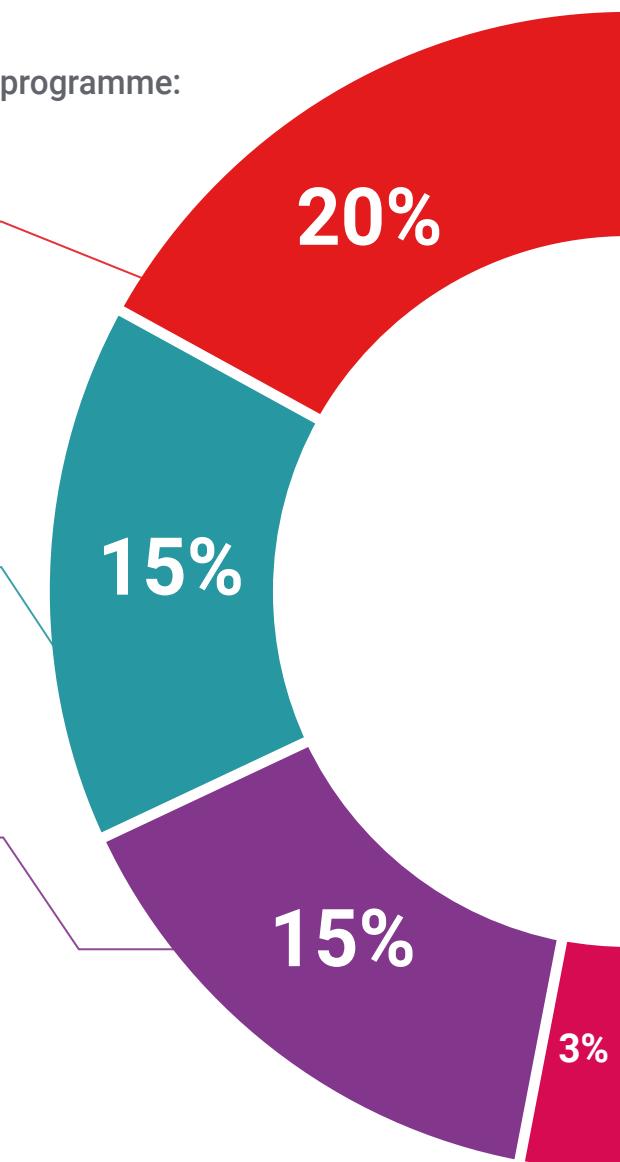
Nous présentons les contenus de manière attrayante et dynamique dans des dossiers multimédias qui incluent de l'audio, des vidéos, des images, des diagrammes et des cartes conceptuelles afin de consolider les connaissances.

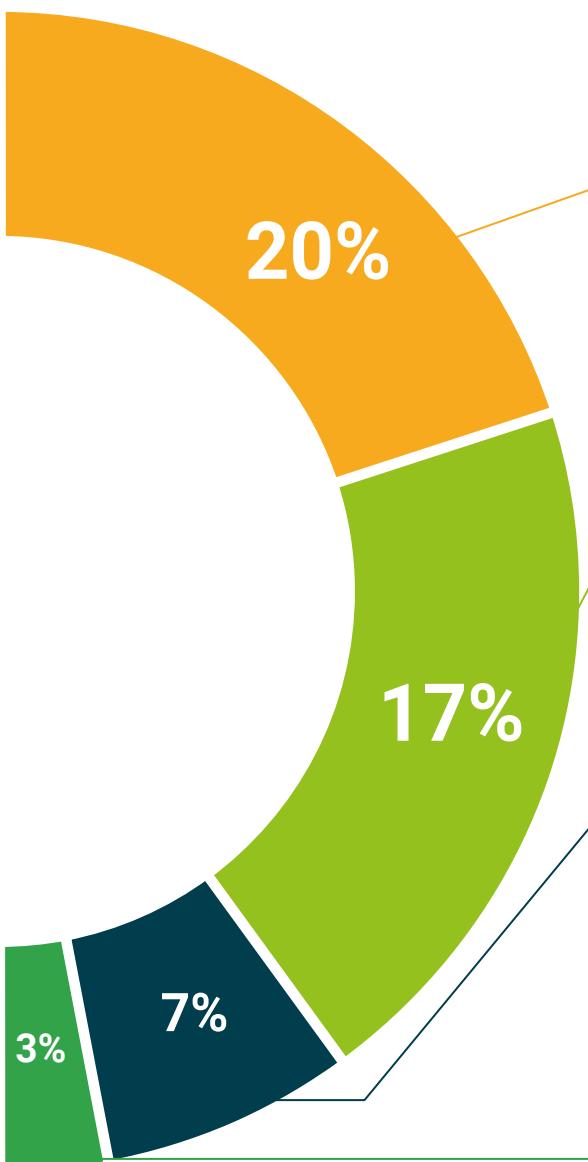
Ce système éducatif unique de présentation de contenu multimédia a été récompensé par Microsoft en tant que "European Success Story".



#### Lectures complémentaires

Articles récents, documents de consensus, guides internationaux, etc... Dans notre bibliothèque virtuelle, vous aurez accès à tout ce dont vous avez besoin pour compléter votre formation.





#### Case Studies

Vous réaliserez une sélection des meilleures *case studies* dans le domaine. Des cas présentés, analysés et encadrés par les meilleurs spécialistes internationaux.



#### Testing & Retesting

Nous évaluons et réévaluons périodiquement vos connaissances tout au long du programme. Nous le faisons sur 3 des 4 niveaux de la Pyramide de Miller.



#### Cours magistraux

Il existe des preuves scientifiques de l'utilité de l'observation par un tiers expert. La méthode *Learning from an Expert* permet au professionnel de renforcer ses connaissances ainsi que sa mémoire, puis lui permet d'avoir davantage confiance en lui concernant la prise de décisions difficiles.



#### Guides d'action rapide

TECH propose les contenus les plus pertinents du programme sous forme de fiches de travail ou de guides d'action rapide. Un moyen synthétique, pratique et efficace pour vous permettre de progresser dans votre apprentissage.



# Corps enseignant

Le corps enseignant de ce programme représente un croisement exceptionnel entre expérience pratique et innovation académique. Composé de professionnels qui non seulement maîtrisent les dernières technologies en matière de développement et d'architecture de Software, mais les appliquent également activement dans des environnements réels, cette équipe garantit une vision actualisée, dynamique et profondément appliquée. De plus, sa capacité à combiner connaissances techniques et réflexion stratégique permet de transformer chaque cours en un espace de création, d'analyse et de résolution de défis concrets. Ainsi, non seulement le contenu est transmis, mais une mentalité prête à diriger l'avenir du Software sous de multiples perspectives est également cultivée.



66

Une équipe enseignante composée de spécialistes renommés en Qualité du Software vous guidera tout au long du programme académique”

## Directeur Invité International

Fort d'une longue carrière de plus de 30 ans dans le secteur technologique, Daniel St. John est un **Ingénieur Informatique** prestigieux hautement spécialisé dans la **Qualité du Logiciel**. Dans cette même optique, il s'est imposé comme un véritable leader dans ce domaine grâce à son approche pragmatique basée sur l'amélioration continue et l'innovation.

Au cours de sa carrière, il a fait partie d'institutions de référence internationale telles que **General Electric Healthcare** dans l'Illinois. Son travail s'est ainsi concentré sur l'optimisation des **infrastructures numériques** des organisations dans le but d'améliorer considérablement **l'expérience des utilisateurs**. Grâce à cela, de nombreux patients ont bénéficié d'une attention plus personnalisée et plus agile, avec un accès plus rapide aux résultats cliniques et au suivi de leur santé. Il a également mis en œuvre des solutions technologiques qui ont permis aux professionnels d'améliorer la **prise de décisions stratégiques** plus éclairées et fondées sur de grands volumes de données.

Il a également combiné ce travail avec la création de projets technologiques de pointe afin de maximiser l'efficacité des processus opérationnels des institutions. À cet égard, il a dirigé la **transformation numérique** de nombreuses entreprises appartenant à différents secteurs. Il a donc mis en place des outils émergents tels que **l'Intelligence Artificielle**, **le Big Data** ou **le Machine Learning** pour automatiser des tâches quotidiennes complexes. Grâce à cela, ces organisations ont réussi à s'adapter rapidement aux tendances du marché et à garantir leur pérennité à long terme.

Il convient de souligner que Daniel St. John a participé en tant qu'intervenant à divers congrès scientifiques à l'échelle mondiale. Il a ainsi partagé ses vastes connaissances dans des domaines tels que l'adoption de **Méthodologies Agiles**, la réalisation de **Tests d'Applications** pour garantir la fiabilité des systèmes ou la mise en œuvre de techniques innovantes de **Blockchain** qui garantissent la protection des données confidentielles.



## M. St. John, Daniel

- 
- Directeur de l'Ingénierie Logicielle chez General Electric Healthcare dans le Wisconsin, États-Unis
  - Chef de l'Ingénierie Logicielle chez Siemens Healthineers, Illinois
  - Directeur de l'Ingénierie Logicielle chez Natus Medical Incorporated, Illinois
  - Ingénieur Senior chez WMS Gaming à Chicago
  - Ingénieur Principal en Logiciels chez Siemens Medical Solutions, Illinois
  - Master en Stratégie et Analyse de Données de l'École Supérieure de Gestion de Lake Forest
  - Diplôme en Informatique de l'Université du Wisconsin-Parkside
  - Membre du Conseil Consultatif de l'Institut de Technologie de l'Illinois
  - Certifications en : Python pour les Sciences des Données, l'Intelligence Artificielle et le Développement, SAFe SCRUM et Project Management

“

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

## Direction



### **M. Molina Molina, Jerónimo**

- Responsable de l'Intelligence Artificielle à Helphone
- AI Engineer & Software Architect en NASSAT Internet par Satellite en Mouvement
- Consultant Senior chez Hexa Engineer
- Introduceur de l'Intelligence Artificielle (ML et CV)
- Expert en Solutions Basées sur l'Intelligence Artificielle, dans le domaine de la *Vision par Ordinateur*, ML/DL et NL
- Expert Universitaire en Création et Développement d'Entreprises à Bancaixa et Fundeun
- Ingénieur en Informatique de l'Université de Alicante
- Master en Intelligence artificielle, Université Catholique de Ávila
- MBA Executive au Forum Européen des Campus d'Affaires



## Professeurs

### **M. Tenrero Morán, Marcos**

- ♦ Ingénieur DevOps chez Allot Communications
- ♦ Manager en Gestion du Cycle de Vie des Applications chez Cegid Meta4
- ♦ Ingénieur en Automatisation QA chez Cegid Meta4
- ♦ Master en Développement d'Applications Professionnelles pour Android par l'Université Galileo. Guatemala
- ♦ Master en Développement de Services Cloud, Node.Js, JavaScript, HTML5 par l'Université Polytechnique de Madrid
- ♦ Développement Web avec Angular-CLI (4), Ionic et Node.Js, Meta4 de l'Université Rey Juan Carlos
- ♦ Diplôme en Ingénierie de ordinateur de l'Université Rey Juan Carlos

### **M. Pi Morell, Oriol**

- ♦ Analyste Fonctionnel à Fihoca
- ♦ Product Owner Hébergement et Email chez CDmon
- ♦ Analyste Fonctionnel et Ingénieur Logiciel à Atmira et CapGemini
- ♦ Enseignant chez Capgemini, Forms Capgemini et Atmira
- ♦ Licence en Génie Technique en Informatique de Gestion, Université Autonome de Barcelone
- ♦ Master en Intelligence artificielle, Université Catholique de Ávila
- ♦ MBA en Gestion et Administration des Entreprises par IMF Smart Education
- ♦ Master en Gestion des Systèmes d'Information par IMF Smart Education
- ♦ Diplôme en Modèles de Conception par l'Université Ouverte de Catalogne (UOC)

**Dr Peralta Martín-Palomino, Arturo**

- CEO et CTO de Prometheus Global Solutions
- CTO chez Korporate Technologies
- CTO d'AI Shephers GmbH
- Consultant et Conseiller Stratégique auprès d'Alliance Medical
- Directeur de la Conception et du Développement chez DocPath
- Docteur en Ingénierie de Informatique de l'Université de Castille - La Manche
- Doctorat en Économie, Commerce et Finances de l'Université Camilo José Cela
- Docteur en Psychologie, Université de Castille - la Manche
- Master en Executive MBA de l'Université Isabel I
- Master en Business and Marketing Management par l'Université Isabel I
- Master Expert en Big Data par Formation Hadoop
- Master en Technologies Avancées de l'Information de l'Université de Castille - la Manche
- Membre de: Groupe de Recherche SMILE

**Mme Martínez Cerrato, Yésica**

- Responsable des Formations Techniques chez Securitas Seguridad España
- Spécialiste en Éducation, Commerce et Marketing
- Product Manager en Sécurité Électronique chez Securitas Seguridad España
- Analyste en Business Intelligence chez Ricopia Technologies
- Technicienne IT et Responsable des Classes Informatiques OTEC à l'Université d'Alcalá de Henares
- Collaboratrice au sein de l'Association ASALUMA
- Diplôme en Ingénierie Électronique des Communications à l'École Polytechnique Supérieure, Université d'Alcalá de Henares



**M. Soto Jiménez, Manuel**

- ◆ Lynx Financial Crime Tech au sein du groupe Santander
- ◆ Diplôme en Ingénierie Informatique de l'Université Autonome de Madrid
- ◆ Diplôme en Mathématiques de l'Université Autonome de Madrid
- ◆ Cours Quantum 101 : Quantum Computing & Quantum Internet Professional Certificate de l'Université Technique de Delft
- ◆ Cours sur le Deep Learning avec TensorFlow par IBM
- ◆ Langages de programmation : Python, R, C, SQL, MongoDB, Matlab, Sage, Cypher, VHDL, Prolog, Javascript, CSS. Langages de Balisage : Markdown, HTML, Latex

“

*Une expérience de formation unique, clé et décisive pour stimuler votre développement professionnel”*

09

# Diplôme

Le Mastère Spécialisé en Qualité du Software garantit, outre la formation la plus rigoureuse et la plus actualisée, l'accès à un diplôme de Mastère Spécialisé délivré par TECH Global University.



66

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

Ce programme vous permettra d'obtenir votre diplôme propre de **Mastère Spécialisé en Qualité du Software** approuvé par **TECH Global University**, la plus grande Université numérique du monde.

**TECH Global University** est une Université Européenne Officielle reconnue publiquement par le Gouvernement d'Andorre ([journal officiel](#)). L'Andorre fait partie de l'Espace Européen de l'Enseignement Supérieur (EEES) depuis 2003. L'EEES est une initiative promue par l'Union européenne qui vise à organiser le cadre international de formation et à harmoniser les systèmes d'enseignement supérieur des pays membres de cet espace. Le projet promeut des valeurs communes, la mise en œuvre d'outils communs et le renforcement de ses mécanismes d'assurance qualité afin d'améliorer la collaboration et la mobilité des étudiants, des chercheurs et des universitaires.



Ce diplôme propre de **TECH Global University** est un programme européen de formation continue et d'actualisation professionnelle qui garantit l'acquisition de compétences dans son domaine de connaissances, conférant une grande valeur curriculaire à l'étudiant qui réussit le programme.

Diplôme: **Mastère Spécialisé en Qualité du Software**

Modalité: **en ligne**

Durée: **12 mois**

Accréditation: **60 ECTS**



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



## Mastère Spécialisé Qualité du Software

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Global University
- » Accréditation: 60 ECTS
- » Horaire: à votre rythme
- » Examens: en ligne

# Mastère Spécialisé

## Qualité du Software

