

# Mastère Hybride

## Ingénierie Avancée du Software



**tech** université  
technologique

## Mastère Hybride Ingénierie Avancée du Software

Modalité: Hybride (en ligne + Stage Pratique)

Durée: 12 mois

Diplôme: TECH Université Technologique

Accès au site web: [www.techtute.com/fr/informatique/mastere-hybride/mastere-hybride-ingenierie-avancee-software](http://www.techtute.com/fr/informatique/mastere-hybride/mastere-hybride-ingenierie-avancee-software)

# Sommaire

01

Présentation

---

Page 4

02

Pourquoi suivre ce Mastère Hybride?

---

Page 8

03

Objectifs

---

Page 12

04

Compétences

---

Page 18

05

Plan d'étude

---

Page 22

06

Stage Pratique

---

Page 38

07

Où puis-je effectuer mon Stage Pratique?

---

Page 44

08

Méthodologie

---

Page 48

09

Diplôme

---

Page 56

# 01

# Présentation

L'Ingénierie Avancée du Software est devenu un élément crucial dans le développement de systèmes complexes et robustes qui sont essentiels à l'économie numérique moderne. Dans un environnement où la demande de logiciels de haute qualité et évolutifs est en constante augmentation, les techniques d'ingénierie du Software avancé permettent aux développeurs de créer des solutions efficaces et durables. Il est donc essentiel que les professionnels de l'Informatique mettent régulièrement à jour leurs connaissances afin d'intégrer dans leur pratique quotidienne les méthodologies les plus innovantes pour améliorer la qualité des logiciels et accélérer le cycle de développement. Dans ce contexte, TECH lance un diplôme universitaire révolutionnaire axé sur les techniques les plus avant-gardistes de l'Ingénierie Avancée du Software.



“

*Grâce à ce Mastère Hybride, vous concevrez les systèmes de Software les plus évolutifs, les plus robustes et les plus faciles à maintenir”*

La qualité et la sécurité du Software sont devenues des éléments essentiels de l'Ingénierie Avancée du Software. À cet égard, un rapport de l'Organisation pour la Coopération et le Développement Économiques révèle que 58% des entreprises connaissent des défaillances de sécurité dues à un manque de pratiques adéquates en matière d'assurance qualité. Les conséquences peuvent être graves, allant de la perte de revenus ou de la perte de confiance des clients à des amendes importantes. D'où l'importance pour les professionnels d'intégrer dans leur pratique les stratégies les plus sophistiquées pour assurer la qualité et la sécurité du software.

Face à ce scénario, TECH présente un Mastère Hybride innovant en Ingénierie Avancée du Software. Il s'agit d'un programme qui combine 1 920 heures du meilleur contenu théorique avec 3 semaines d'expérience pratique dans une entreprise leader dans ce domaine. Composé de 10 modules complets, l'itinéraire académique approfondira des sujets tels que l'Ingénierie des Exigences, la Conception d'Applications Web ou l'Audit des Systèmes d'Information. En outre, le programme d'études fournira aux étudiants des techniques de pointe pour la protection des systèmes et le développement de codes hautement sécurisés. De cette manière, les diplômés acquerront des compétences avancées pour analyser des problèmes de software complexes et développer des solutions efficaces et innovantes.

D'autre part, ce diplôme universitaire prévoit que les diplômés effectuent un stage pratique dans une institution prestigieuse dans le domaine de l'Ingénierie Avancée du Software. Ainsi, les étudiants participeront activement aux projets en cours de développement à ce moment-là. Il convient de noter qu'un tuteur spécialisé guidera les étudiants pendant l'expérience académique, garantissant la réalisation d'un plan d'activités qui leur permettra d'améliorer leurs compétences de manière exponentielle et sur la base des exigences de la demande actuelle du marché du travail.

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

- ♦ Le développement de 100 cas pratiques présentés par des experts en Ingénierie Informatique
- ♦ Les contenus graphiques, schématiques et éminemment pratiques avec lesquels elles sont conçues fournissent des informations essentielles sur les outils et les techniques indispensables à la pratique professionnelle
- ♦ Les exercices pratiques où le processus d'auto-évaluation peut être réalisé afin d'améliorer l'apprentissage
- ♦ Il met l'accent sur les méthodologies innovantes
- ♦ Tout cela sera complété par des cours théoriques, des questions à l'expert, des forums de discussion sur des sujets controversés et un travail de réflexion individuel
- ♦ Les contenus sont disponibles à partir de tout appareil fixe ou portable doté d'une connexion internet
- ♦ En outre, vous pourrez effectuer un stage dans l'une des meilleures entreprises du secteur



*Vous dirigerez des projets de Développement de Software, de la conception à la mise en œuvre et à l'évaluation”*

“

*Vous effectuerez un stage intensif de 3 semaines dans une institution de référence dans le domaine de l'Ingénierie Avancée du Software”*

Dans cette proposition de Mastère, de nature professionnalisante et de modalité d'apprentissage hybride, le programme est destiné à mettre à jour les professionnels de l'Informatique. Le contenu est basé sur les dernières données scientifiques, et orientés de manière didactique pour intégrer les connaissances théoriques dans la pratique informatique, et les éléments théoriques et pratiques faciliteront la mise à jour des connaissances.

Grâce à son contenu multimédia développé avec les dernières technologies éducatives, ils permettront au professionnel de l'Informatique un apprentissage situé et contextuel, c'est-à-dire un environnement simulé qui fournira un apprentissage immersif programmé pour s'entraîner dans des situations réelles. La conception de ce programme est basée sur l'Apprentissage Par les Problèmes, grâce auquel le médecin devra essayer de résoudre les différentes situations de pratique professionnelle qui se présentent tout au long du programme. Pour ce faire, l'étudiant sera assisté d'un innovant système de vidéos interactives, créé par des experts reconnus.

*Vous mettrez à jour vos connaissances en Ingénierie Avancée du Software grâce à un contenu multimédia innovant.*

*TECH est une université à la pointe de la technologie, qui met toutes ses ressources à votre disposition pour vous aider à réussir en tant qu'Informaticien.*



# 02

## Pourquoi suivre ce Mastère Hybride?

L'une des principales priorités des entreprises est d'innover et de rester compétitives sur le marché mondial. C'est pourquoi les organisations cherchent à recruter des informaticiens hautement spécialisés en Ingénierie Avancée du Software, capables de développer de nouvelles applications qui leur procureront des avantages concurrentiels. Pour profiter de ces opportunités, les professionnels doivent se tenir au courant des derniers développements dans ce domaine. C'est dans cette optique que TECH a créé un programme académique unique et novateur dans le paysage éducatif actuel, qui permettra aux spécialistes de s'immerger dans un environnement de travail réel et d'appliquer les procédures et techniques les plus récentes dans le domaine de l'Ingénierie Avancée du Software.



```
manager.create(this, TOOLTIP_ID_CONTACT)  
viewById(R.id.btnContact), TooltipManager.GRAVITY.BOTTOM)  
TooltipManager.ClosePolicy.TOUCH_OUTSIDE, 3000)  
activateDelay(300)  
fitToScreen(true)  
withStyleId(R.style.TooltipLayoutStyle)  
text(R.string.label_tips_contact);  
mTooltipContact.show();  
I  
)  
@Override  
public void onBackPressed() {  
    super.onBackPressed();  
}  
private String validationRegisterForm() {  
    String errorMessage = null;
```

“

*Vous effectuerez votre séjour pratique dans une entité de référence, où vous analyserez les dernières avancées en Ingénierie Avancée du Software”*

### 1. Actualisation des technologies les plus récentes

Les nouvelles technologies ont eu un impact significatif sur le domaine de l'Ingénierie Avancée du Software, en améliorant l'efficacité, la qualité et la capacité d'innovation dans le développement de software. Les tests automatisés, qui permettent aux experts de détecter et de corriger rapidement les erreurs, en sont un exemple. Dans le but de rapprocher ces outils des spécialistes, TECH présente cette Formation Pratique qui permet aux étudiants d'entrer dans un environnement de travail de pointe, où ils auront accès à la technologie de dernière génération dans ce domaine.

### 2. Exploiter l'expertise des meilleurs spécialistes

Tout au long de la Formation Pratique, une équipe d'experts en Ingénierie Avancé du Software guidera les étudiants, les aidant à tirer le meilleur parti de cette expérience académique. En ce sens, ces professionnels transmettront aux diplômés toutes les connaissances dont ils ont besoin pour propulser leur carrière d'informaticien au sommet.

### 3. Accéder à des environnements professionnels de premier ordre

L'objectif principal de TECH est d'offrir des programmes universitaires de haute qualité et accessibles à tous. C'est pourquoi TECH sélectionne avec soin les centres où les diplômés effectueront leur stage. Cela permet aux informaticiens d'avoir accès à des institutions de premier plan dans le domaine de l'Ingénierie Avancée du Software. Ainsi, les diplômés pourront faire l'expérience directe du travail quotidien dans un environnement exigeant et rigoureux, en appliquant toujours les techniques et les méthodologies les plus avancées.



#### 4. Combiner les meilleures théories avec les pratiques les plus modernes

Dans le paysage académique actuel, il est courant de trouver des programmes universitaires qui n'offrent qu'un contenu théorique, laissant de côté l'importance de la pratique pour que les étudiants puissent appliquer leurs connaissances dans des situations de travail réelles. En revanche, TECH présente un modèle d'apprentissage entièrement pratique, qui permet aux professionnels de l'Informatique d'acquérir une expérience concrète et de relever les défis réels qui peuvent se présenter au cours de leur carrière professionnelle.

#### 5. Élargir les frontières de la connaissance

TECH offre aux diplômés la possibilité de suivre ce Mastère Hybride dans des institutions internationales prestigieuses. Cela permet aux informaticiens d'élargir leurs horizons et de mettre à jour leurs connaissances avec les meilleurs professionnels travaillant dans des entreprises de haut niveau. Il s'agit d'une opportunité exceptionnelle que seule TECH, la plus grande université numérique du monde, peut offrir.

“

*Vous serez en immersion totale  
dans le centre de votre choix”*

# 03 Objectifs

À l'issue de ce Mastère Hybride, les informaticiens auront une compréhension globale de la conception, de l'architecture, de la mise en œuvre et de la maintenance de systèmes de Software complexes. En ce sens, les diplômés appliqueront des méthodologies de développement de Software à la fois agiles et traditionnelles afin de gérer des projets de manière efficace. En outre, les professionnels utiliseront des indicateurs pour mesurer et améliorer la qualité des applications.



“

*Incorporez dans votre pratique les techniques de développement sécurisé les plus innovantes pour protéger les systèmes de Software contre les menaces et les vulnérabilités”*



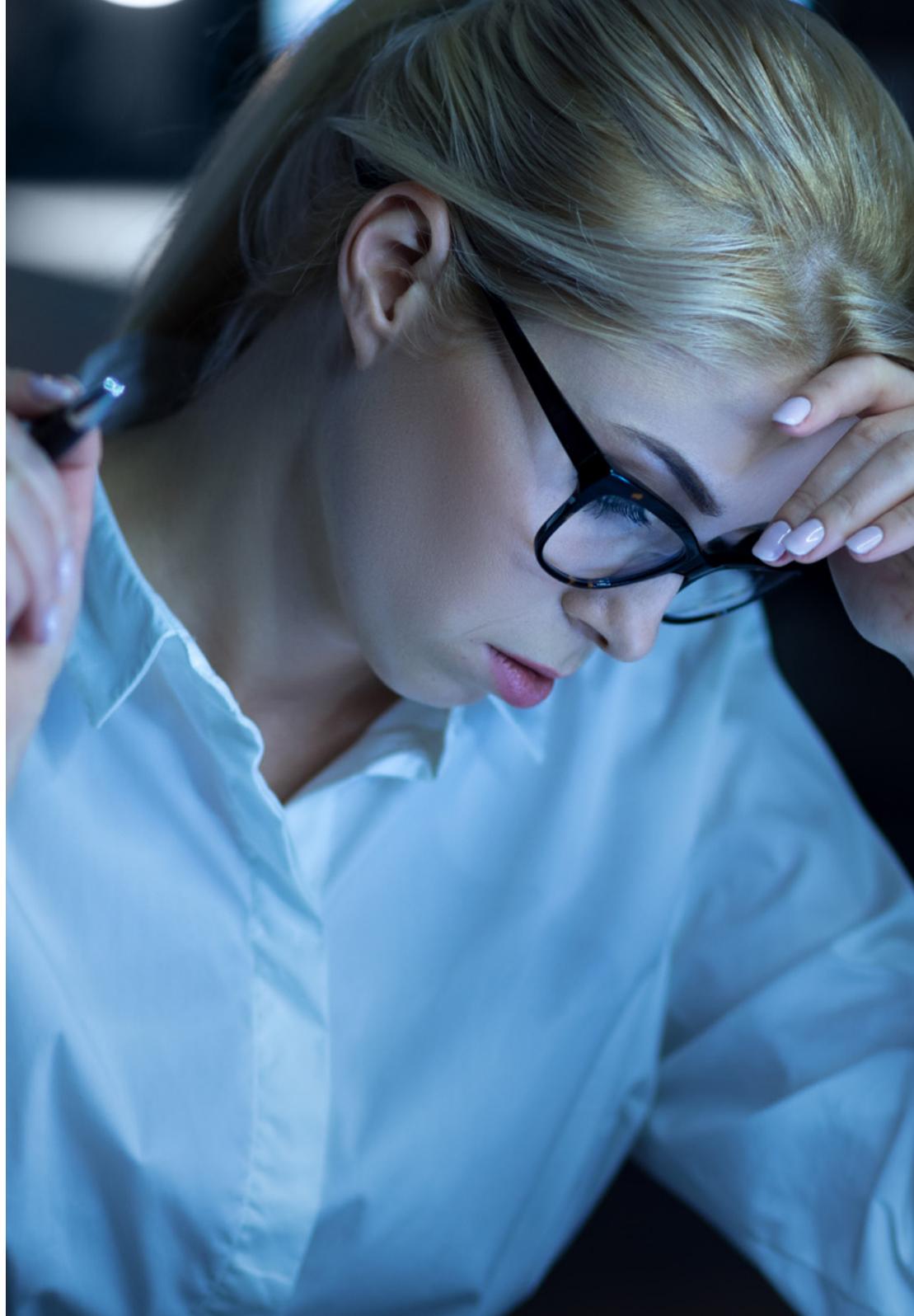
## Objectif général

---

- ♦ Ce Mastère Hybride en Ingénierie Avancée du Software fournira aux professionnels de l'informatique des compétences avancées pour concevoir des architectures logicielles évolutives, robustes et maintenables en utilisant des schémas de conception et des principes d'architecture de Software. En même temps, les étudiants appliqueront des techniques de modélisation et de simulation pour prédire et résoudre efficacement les problèmes potentiels dans les systèmes de Software

“

*Vous apprendrez à travers des cas réels et en résolvant des situations complexes dans des environnements d'apprentissage simulés”*





## Objectifs spécifiques

---

### Module 1. Ingénierie Software

- ♦ Poser les bases de l'Ingénierie du Software et de la modélisation, en apprenant les principaux processus et concepts
- ♦ Comprendre le processus Logiciel et les différents modèles pour son développement, y compris les technologies agiles

### Module 2. Ingénierie Avancée du Software

- ♦ Découvrez en profondeur les différentes méthodologies agiles utilisées dans l'Ingénierie du Software
- ♦ Apprendre à développer les techniques en utilisant Scrum, la programmation extrême et de développement logiciel basées sur la réutilisation
- ♦ Comprendre les concepts et les processus de conception de software, en apprenant également sur la conception de l'architecture et sur la conception au niveau des composants et basée sur des modèles
- ♦ Introduire le concept de DevOps et quelles sont ses principales pratiques
- ♦ Apprendre à tester le logiciel, avec des méthodologies telles que *Test Driven Development*, *Acceptance Test Driven Development*, *Behavior Driven Development*, BDD et *Cucumber*
- ♦ Comprendre les différents modèles d'architectures de systèmes et conception de logiciels, ainsi que l'architecture des applications en nuage

### Module 3. Ingénierie des exigences

- ♦ Comprendre l'ingénierie des exigences, leur développement, leur élaboration, leur négociation et leur validation
- ♦ Apprenez la modélisation des exigences et les différents éléments tels que les scénarios, les informations, les classes d'analyse, le flux, le comportement et les modèles
- ♦ Comprendre l'importance de l'ingénierie des besoins dans le processus de développement de software
- ♦ Apprendre à réaliser des analyses et à documenter correctement les exigences
- ♦ Approfondir les sources d'exigences et les techniques d'appel d'offres, car elles sont un élément essentiel du processus
- ♦ Comprendre les processus de validation et de négociation des exigences, ainsi que la modélisation et la gestion des exigences

### Module 4. Processus d'Ingénierie Software

- ♦ Approfondir l'amélioration du processus de développement de logiciels et de la qualité des logiciels en utilisant les normes ISO/IEC
- ♦ Comprendre et appliquer le prototypage comme partie essentielle du processus de développement
- ♦ Connaître le cadre de référence de l'Ingénierie de software et la norme ISO/IEC 12207
- ♦ Apprendre les caractéristiques du processus unifié de développement de logiciels et de planification dans le contexte du développement de software agile
- ♦ Connaître les différents styles de conception de logiciels distribués et les architectures logicielles orientées services
- ♦ Apprendre les concepts essentiels dans la conception d'interfaces graphiques utilisateur

### Module 5. Qualité et audit des systèmes d'information

- ♦ Approfondir les stratégies et techniques de test des logiciels, les facteurs de qualité des logiciels et les différentes mesures utilisées
- ♦ Acquérir les connaissances essentielles des systèmes de gestion de la sécurité informatique
- ♦ Introduire les concepts de propriété intellectuelle dans les systèmes de gestion de l'information
- ♦ Préparer les étudiants à la création de plans de continuité des activités et de reprise après un sinistre
- ♦ Apprendre à planifier la gestion de la sécurité et à gérer les principaux mécanismes de protection des ressources et des informations
- ♦ Connaître les différents types d'audits et connaître le processus d'audit informatique

### Module 6. Intégration des systèmes

- ♦ Acquérir les concepts essentiels liés aux systèmes d'information dans l'entreprise et identifier les opportunités et les besoins des systèmes d'information dans l'entreprise
- ♦ Connaître les bases de la *Business Intelligence*, ses stratégies et sa mise en oeuvre, ainsi que le présent et l'avenir du BI
- ♦ Comprendre le fonctionnement des systèmes de gestion intégrée des ressources de l'entreprise
- ♦ Comprendre la transformation numérique du point de vue de l'innovation commerciale, de la gestion financière et de la production, du marketing et de la gestion des ressources humaines

### Module 7. Réutilisation de software

- ♦ Présentation de la stratégie de réutilisation software
- ♦ Apprendre les différents schémas liés à la réutilisation des logiciels, qu'ils soient de conception, de création, de structure ou de comportement
- ♦ Introduire le concept de *framework*, et en connaître les principaux types, tels que ceux destinés à la conception d'interfaces graphiques utilisateur, au développement d'applications Web et à la gestion de la persistance d'objets dans les bases de données
- ♦ Comprendre le fonctionnement du modèle largement utilisé actuellement de Model Vista Controller (MVC)

### Module 8. Services informatiques

- ♦ Formation à la prise de décision d'investissement dans les TIC et à la planification des systèmes d'information
- ♦ Connaître les objectifs de contrôle pour les technologies et les informations connexes (COBIT)
- ♦ Apprendre le fonctionnement de la bibliothèque d'infrastructure des technologies de l'information (ITIL), les stratégies, la conception des services, les transitions et les opérations
- ♦ Approfondir le système de gestion des services, en connaissant les principes de base de UNE-ISO/IEC 20000-1, la structure de la série de normes ISO/IEC 20000 et les exigences du Système de Gestion des Services (SGS)
- ♦ Comprendre le fonctionnement des systèmes et des technologies de l'information, leurs composants, leurs classifications, leurs architectures et leurs modalités d'intégration
- ♦ Apprendre la norme ISO/IEC 12207, l'analyse, la conception, la mise en oeuvre et l'acceptation des systèmes d'information



### Module 9. Sécurité des systèmes d'information

- ♦ Apprendre à élaborer un calendrier de gestion du temps, de développement budgétaire et de gestion des risques
- ♦ Analyser la nature des attaques réseau et les différents types d'architectures de sécurité
- ♦ Comprendre les différentes techniques de protection des systèmes et de développement de codes sécurisés
- ♦ Comprendre les composantes essentielles des *botnets* et des spam, ainsi que des malware et des codes malveillants
- ♦ Jeter les bases de l'analyse médico-légale dans le monde du logiciel et de l'audit informatique
- ♦ Comprendre les bases de la cryptographie symétrique et de la cryptographie asymétrique, ainsi que leurs principaux algorithmes

### Module 10. Gestion de projets

- ♦ Comprendre le fonctionnement de la gestion de la qualité dans les projets, y compris la planification, l'assurance, le contrôle, les concepts statistiques et les outils disponibles
- ♦ Comprendre le fonctionnement des processus d'approvisionnement, d'exécution, de surveillance, de contrôle et de clôture d'un projet
- ♦ Acquérir les connaissances essentielles relatives à la responsabilité professionnelle découlant de la gestion de projets
- ♦ Connaître les concepts fondamentaux et le cycle de vie de la gestion de projets

# 04

# Compétences

Grâce à ce diplôme universitaire, les professionnels de l'informatique acquerront des compétences avancées pour concevoir des systèmes logiciels évolutifs, maintenables et robustes en utilisant les principes de l'architecture de conception. En conséquence, les diplômés seront hautement qualifiés pour appliquer les méthodologies agiles (telles que Scrum ou Kanban) dans le cycle de vie du développement de software. En outre, les experts mettront en œuvre des processus d'assurance qualité afin de garantir la qualité des applications par le biais de revues de code des applications par des revues de code, des audits et des tests approfondis.



“

*Vous utiliserez des Frameworks pour maintenir la cohérence dans la conception et la mise en œuvre du Software”*



## Compétences générales

---

- ◆ Répondre aux besoins actuels dans le domaine de l'Ingénierie Avancée du Software
- ◆ Avoir la maîtrise des différents systèmes de travail dans l'Ingénierie Avancée du Software

“

*Vous disposerez d'une bibliothèque pleine de ressources pédagogiques disponibles 24 heures sur 24 et d'un matériel qui se distingue par sa qualité”*





## Compétences spécifiques

---

- ♦ Découvrir en profondeur toutes les facettes de l'interaction homme-ordinateur et la façon dont les développements informatiques sont impliqués
- ♦ Être solvable dans l'utilisation des bases de données
- ♦ Développer différents types d'applications en réseau
- ♦ Travailler comme ingénieur logiciel
- ♦ Contrôler l'utilisation des bases de données avancées
- ♦ Planification avancée
- ♦ Savoir comment le logiciel est réutilisé
- ♦ Création d'applications et d'interfaces réseau



*Vous combinerez la théorie et la pratique professionnelle dans le cadre d'une approche pédagogique exigeante et enrichissante”*

# 05

## Plan d'étude

Le matériel pédagogique qui compose ce Mastère Hybride a été préparé par une équipe de professionnels hautement spécialisés dans l'Ingénierie Avancée du Software. Ils ont ainsi développé un programme qui se distingue à la fois par son excellente qualité et par son adéquation avec les exigences du marché du travail actuel. Composé de 10 modules spécialisés, le programme se penchera sur les dernières innovations dans des domaines tels que la Modélisation des Exigences, l'Architecture du Software ou la Conception de l'Interface Utilisateur Graphique. En outre, le programme fournira aux diplômés les techniques de pointe pour la maintenance du software.

manager.js  
main.js  
readme.rst  
sequences.js  
sessions.js  
settings.js  
tasks.js  
templates

56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71

```
sel  
ret  
}  
if (e  
e.s  
e.p  
sel  
sel  
ret  
}  
//up/  
if (e  
e.p  
e.s  
if  
/  
i
```

```
f.deactivate(true);  
return; // this is a modified line  
  
.keyCode == 13) {  
  topImmediatePropagation();  
  reventDefault();  
  f.search();  
  f.deactivate();  
  return;  
  
down  
.keyCode == 38 || e.keyCode == 40) {  
  reventDefault();  
  topImmediatePropagation();  
  (e.keyCode == 38) { // up  
    / show previous search quer  
    f (hist.currentIndex ==  
    hist.temporaryQuer  
    // skip previous  
    if (hist.t  
    hist  
  }  
}
```

“

*Vous maîtriserez les Méthodologies Agiles pour améliorer l'efficacité, la flexibilité et la réactivité des équipes de développement face aux changements”*

## Module 1. Ingénierie Software

- 1.1. Introduction au Génie Logiciel et à la modélisation
  - 1.1.1. La nature des logiciels
  - 1.1.2. La nature unique des webApps
  - 1.1.3. Ingénierie Software
  - 1.1.4. Le processus logiciel
  - 1.1.5. La pratique du Génie Logiciel
  - 1.1.6. Mythes sur les logiciels
  - 1.1.7. Comment tout commence
  - 1.1.8. Concepts orientés objet
  - 1.1.9. Introduction à UML
- 1.2. Le processus logiciel
  - 1.2.1. Un modèle général de processus
  - 1.2.2. Modèles de processus prescriptifs
  - 1.2.3. Modèles de processus spécialisés
  - 1.2.4. Le processus unifié
  - 1.2.5. Modèles de processus personnels et d'équipe
  - 1.2.6. Qu'est-ce que l'agilité?
  - 1.2.7. Qu'est-ce qu'un processus agile?
  - 1.2.8. Scrum
  - 1.2.9. Boîte à outils du processus Agile
- 1.3. Principes guidant la pratique du Génie Logiciel
  - 1.3.1. Principes guidant le processus
  - 1.3.2. Principes guidant la pratique
  - 1.3.3. Principes de la communication
  - 1.3.4. Principes de planification
  - 1.3.5. Principes de modélisation
  - 1.3.6. Principes de construction
  - 1.3.7. Principes de déploiement



- 1.4. Comprendre les besoins
  - 1.4.1. Ingénierie des exigences
  - 1.4.2. Établir la base
  - 1.4.3. Détermination des besoins
  - 1.4.4. Développer des cas d'utilisation
  - 1.4.5. Élaboration du modèle d'exigences
  - 1.4.6. Négociation des exigences
  - 1.4.7. Validation des exigences
- 1.5. Modélisation des exigences I: scénarios, classes d'information et d'analyse
  - 1.5.1. Analyse des besoins
  - 1.5.2. Modélisation basée sur des scénarios
  - 1.5.3. Modèles UML fournissant le cas d'utilisation
  - 1.5.4. Concepts de modélisation des données
  - 1.5.5. Modélisation basée sur les classes
  - 1.5.6. Diagrammes de classes
- 1.6. Modélisation des exigences II: flux, comportement et modèles
  - 1.6.1. Stratégies de modélisation des exigences
  - 1.6.2. Modélisation orientée flux
  - 1.6.3. Diagrammes d'état
  - 1.6.4. Création d'un modèle comportemental
  - 1.6.5. Diagrammes de séquence
  - 1.6.6. Diagrammes de communication
  - 1.6.7. Modèles pour la modélisation des exigences
- 1.7. Concepts de design
  - 1.7.1. Conception dans le contexte de l'Ingénierie Logicielle
  - 1.7.2. Le processus de conception
  - 1.7.3. Concepts de design
  - 1.7.4. Concepts de conception orientée objet
  - 1.7.5. Le modèle de conception
- 1.8. Architecture de conception
  - 1.8.1. Architecture logicielle
  - 1.8.2. Genres architecturaux
  - 1.8.3. Styles architecturaux
  - 1.8.4. Conception architecturale
  - 1.8.5. Évolution des conceptions alternatives de l'architecture
  - 1.8.6. Cartographie de l'architecture à l'aide du flux de données
- 1.9. Conception au niveau des composants et des modèles
  - 1.9.1. Qu'est-ce qu'un composant?
  - 1.9.2. Conception de composants basée sur les classes
  - 1.9.3. Réalisation de la conception au niveau des composants
  - 1.9.4. Conception traditionnelle des composants
  - 1.9.5. Développement basé sur les composants
  - 1.9.6. Modèles de conception
  - 1.9.7. Conception de logiciels basée sur des modèles
  - 1.9.8. Modèles architecturaux
  - 1.9.9. Patrons de conception au niveau des composants
  - 1.9.10. Modèles de conception d'interface utilisateur
- 1.10. Qualité des logiciels et gestion de projet
  - 1.10.1. Qualité
  - 1.10.2. Qualité des logiciels
  - 1.10.3. Le dilemme de la qualité des logiciels
  - 1.10.4. Atteindre la qualité des logiciels
  - 1.10.5. Assurance qualité des logiciels
  - 1.10.6. Le spectre de la gestion
  - 1.10.7. Personnel
  - 1.10.8. Le produit
  - 1.10.9. Le processus
  - 1.10.10. Le projet
  - 1.10.11. Principes et pratiques

## Module 2. Ingénierie Avancée du Software

- 2.1. Introduction aux méthodologies Agiles
  - 2.1.1. Modèles de processus et méthodologies
  - 2.1.2. Agilité et processus agiles
  - 2.1.3. Manifeste agile
  - 2.1.4. Quelques méthodologies agiles
  - 2.1.5. Agile vs. Traditionnel
- 2.2. Scrum
  - 2.2.1. Origines et philosophie de Scrum
  - 2.2.2. Valeurs Scrum
  - 2.2.3. Flux de processus Scrum
  - 2.2.4. Les rôles Scrum
  - 2.2.5. Les artefacts de Scrum
  - 2.2.6. Les événements Scrum
  - 2.2.7. Histoires d'utilisateur
  - 2.2.8. Extensions Scrum
  - 2.2.9. Estimations rapides
  - 2.2.10. Escalade de Scrum
- 2.3. Programmation extrême
  - 2.3.1. Justification et présentation de XP
  - 2.3.2. Le cycle de vie au XP
  - 2.3.3. Les cinq valeurs de base
  - 2.3.4. Les douze pratiques de base de XP
  - 2.3.5. Rôles des participants
  - 2.3.6. XP Industriel
  - 2.3.7. Évaluation critique de XP
- 2.4. Développement de logiciels basés sur la réutilisation
  - 2.4.1. Réutilisation de software
  - 2.4.2. Niveaux de réutilisation du code
  - 2.4.3. Techniques spécifiques de réutilisation
  - 2.4.4. Développement basé sur les composants
  - 2.4.5. Avantages et problèmes de réutilisation
  - 2.4.6. Planification de la réutilisation
- 2.5. Modèles d'architecture système et de conception logicielle
  - 2.5.1. Conception architecturale
  - 2.5.2. Modèles architecturaux généraux
  - 2.5.3. Architectures tolérantes aux pannes
  - 2.5.4. Architectures de systèmes distribués
  - 2.5.5. Modèles de conception
  - 2.5.6. Modèles Gamma
  - 2.5.7. Modèles de conception d'interaction
- 2.6. Architecture d'application cloud
  - 2.6.1. Principes fondamentaux de Cloud Computing
  - 2.6.2. Qualité d'application cloud
  - 2.6.3. Styles d'architecture
  - 2.6.4. Modèles de conception
- 2.7. Tests logiciels: TDD, ATDD et BDD
  - 2.7.1. Vérification et validation du logiciel
  - 2.7.2. Tests logiciels
  - 2.7.3. Test Driven Development (TDD)
  - 2.7.4. Acceptance Test Driven Development (ATDD)
  - 2.7.5. Behavior Driven Development (BDD)
  - 2.7.6. BDD et Cucumber
- 2.8. L'amélioration du processus logiciel
  - 2.8.1. L'amélioration du processus logiciel
  - 2.8.2. Le processus d'amélioration de processus
  - 2.8.3. Modèles de maturité
  - 2.8.4. Le modèle CMMI
  - 2.8.5. CMMI V2.0
  - 2.8.6. CMMI et Agile

- 2.9. Qualité des systèmes informatiques: SQuaRE
  - 2.9.1. Qualité des logiciels
  - 2.9.2. Modèles de qualité des produit logiciels
  - 2.9.3. Famille ISO/IEC 25000
  - 2.9.4. ISO/IEC 25010: modèle et caractéristiques de qualité
  - 2.9.5. ISO/IEC 25012: qualité des données
  - 2.9.6. ISO/IEC 25020: mesure de la qualité du logiciel
  - 2.9.7. ISO/IEC 25022, 25023 et 25024: mesures de qualité des logiciels et des données
  - 2.9.8. ISO/IEC 25040: évaluation du logiciel
  - 2.9.9. Processus de certification
- 2.10. Présentation de DevOps
  - 2.10.1. Concept DevOps
  - 2.10.2. Principales pratiques

### Module 3. Ingénierie des exigences

- 3.1. Introduction à l'Ingénierie des Exigences
  - 3.1.1. L'importance des exigences
  - 3.1.2. Concept d'exigence
  - 3.1.3. Dimensions des exigences
  - 3.1.4. Niveaux et types d'exigences
  - 3.1.5. Caractéristiques des exigences
  - 3.1.6. Ingénierie des exigences
  - 3.1.7. Le processus d'Ingénierie des Exigences
  - 3.1.8. *Frameworks* pour l'ingénierie des exigences
  - 3.1.9. Meilleures pratiques en matière d'ingénierie des exigences
  - 3.1.10. L'analyste d'affaires
- 3.2. Les sources des besoins
  - 3.2.1. Le réseau des exigences
  - 3.2.2. Les *stakeholders*
  - 3.2.3. Les exigences commerciales
  - 3.2.4. Document sur la vision et le champ d'application

- 3.3. Techniques d'élicitation des besoins
  - 3.3.1. Détermination des besoins
  - 3.3.2. Problèmes liés à la détermination des besoins
  - 3.3.3. Contextes de découverte
  - 3.3.4. Interviews
  - 3.3.5. Observation et "apprentissage"
  - 3.3.6. Ethnographie
  - 3.3.7. Workshops
  - 3.3.8. *Focus groups*
  - 3.3.9. Questionnaires
  - 3.3.10. *Brainstorming* et techniques de création
  - 3.3.11. Médias du groupe
  - 3.3.12. Analyse des interfaces du système
  - 3.3.13. Analyse de documents et "archéologie"
  - 3.3.14. Cas d'utilisation et scénario
  - 3.3.15. Prototypes
  - 3.3.16. Ingénierie inverse
  - 3.3.17. Réutilisation des exigences
  - 3.3.18. Bonnes pratiques en matière d'appels d'offres
- 3.4. Besoins des utilisateurs
  - 3.4.1. Personnes
  - 3.4.2. Cas d'utilisation et histoires d'utilisateurs
  - 3.4.3. Scénario
  - 3.4.5. Types de scénarios
  - 3.4.6. Comment découvrir des scénarios
- 3.5. Techniques de prototypage
  - 3.5.1. Prototypage
  - 3.5.2. Prototypes en fonction de leur portée
  - 3.5.3. Prototypes en fonction de leur temporalité
  - 3.5.4. La fidélité d'un prototype
  - 3.5.5. Prototypes d'interface utilisateur
  - 3.5.6. Évaluation des prototypes
- 3.6. Analyse des besoins
  - 3.6.1. Analyse des besoins
  - 3.6.2. Bonnes pratiques en matière d'analyse des exigences
  - 3.6.3. Dictionnaire de données
  - 3.6.4. Hiérarchisation des besoins
- 3.7. Documentation des exigences
  - 3.7.1. Le document de spécification des exigences
  - 3.7.2. Structure et contenu d'un BSS
  - 3.7.3. Documentation en langage naturel
  - 3.7.4. EARS: Easy Approach to Requirements Syntax
  - 3.7.5. Exigences non fonctionnelles
  - 3.7.6. Attributs et modèles de table
  - 3.7.7. Bonne pratique de spécification
- 3.8. Validation et négociation des exigences
  - 3.8.1. Validation des exigences
  - 3.8.2. Techniques de validation des exigences
  - 3.8.3. Négociation des exigences
- 3.9. Modélisation et gestion des exigences
  - 3.9.1. Modélisation des besoins
  - 3.9.2. Le point de vue de l'utilisateur
  - 3.9.3. Perspective des données
  - 3.9.4. La perspective fonctionnelle ou axée sur le flux
  - 3.9.5. La perspective comportementale
  - 3.9.6. Volatilité des exigences
  - 3.9.7. Processus de gestion des exigences
  - 3.9.8. Outils de gestion des exigences
  - 3.9.9. Bonnes pratiques en matière de gestion des exigences
- 3.10. Systèmes critiques et spécification formelle
  - 3.10.1. Systèmes critiques
  - 3.10.2. Spécification axée sur le risque
  - 3.10.3. Spécification formelle

## Module 4. Processus d'Ingénierie Software

- 4.1. Cadre de l'Ingénierie logicielle
  - 4.1.1. Caractéristiques des logiciels
  - 4.1.2. Les principaux processus de l'Ingénierie Software
  - 4.1.3. Modèles de processus de développement de logiciels
  - 4.1.4. Cadre de référence normalisé pour le processus de développement de software: la norme ISO/CEI 12207
- 4.2. Processus unifié de développement de logiciels
  - 4.2.1. Processus Unifié
  - 4.2.2. Les dimensions du Processus Unifié
  - 4.2.3. Processus de développement basé sur les cas d'utilisation
  - 4.2.4. Flux de travail fondamentaux du Processus Unifié
- 4.3. La planification dans le contexte du développement logiciel agile
  - 4.3.1. Caractéristiques du développement logiciel agile
  - 4.3.2. Différents horizons de planification dans le développement agile
  - 4.3.3. Cadre de développement agile Scrum et planification des horizons temporels
  - 4.3.4. User stories comme unité de planification et d'estimation
  - 4.3.5. Techniques courantes de calcul d'une estimation
  - 4.3.6. Échelles d'interprétation des estimations
  - 4.3.7. *Planning poker*
  - 4.3.8. Types courants de planification: planification des livraisons et planification des itérations
- 4.4. Styles de conception de logiciels distribués et architectures logicielles orientées services
  - 4.4.1. Modèles de communication dans les systèmes logiciels distribués
  - 4.4.2. Couche intermédiaire
  - 4.4.3. Modèles architecturaux pour les systèmes distribués
  - 4.4.4. Processus général de conception de services logiciels
  - 4.4.5. Aspects de la conception de services logiciels
  - 4.4.6. Composition des services
  - 4.4.7. Architecture des services web
  - 4.4.8. Composants d'infrastructure et SOA
- 4.5. Introduction au développement logiciel guidé par le modèle
  - 4.5.1. Le concept de modèle
  - 4.5.2. Développement logiciel guidé par le modèle
  - 4.5.3. MDA cadre de développement guidé par le modèle
  - 4.5.4. Éléments d'un modèle de transformation
- 4.6. Conception d'interfaces utilisateur graphiques
  - 4.6.1. Principes de conception d'interfaces utilisateur
  - 4.6.2. Modèles de conception architecturale pour les systèmes interactifs: Modèle Vue Contrôleur Vue (MVC)
  - 4.6.3. Expérience utilisateur (UX User Experience)
  - 4.6.4. Conception centrée sur l'utilisateur
  - 4.6.5. Processus d'analyse et de conception d'une interface utilisateur graphique
  - 4.6.6. Utilisabilité d'interfaces utilisateurs
  - 4.6.7. Accessibilité des interfaces utilisateurs
- 4.7. Conception d'applications web
  - 4.7.1. Caractéristiques des applications web
  - 4.7.2. Interface utilisateur d'une application web
  - 4.7.3. Conception de la navigation
  - 4.7.4. Protocole d'interaction de base pour les applications web
  - 4.7.5. Styles d'architecture pour les applications web
- 4.8. Stratégies et techniques de test des logiciels et facteurs de qualité des logiciels
  - 4.8.1. Stratégies d'essai
  - 4.8.2. Designs de cas d'essai
  - 4.8.3. Rapport qualité-prix
  - 4.8.4. Modèles de qualité
  - 4.8.5. Famille de normes ISO/IEC 25000 (SQuaRE)
  - 4.8.6. Modèle de qualité du produit (ISO 2501n)
  - 4.8.7. Modèles de qualité des données (ISO 2501n)
  - 4.8.8. Gestion de la qualité des logiciels

- 4.9. Introduction aux métriques de l'ingénierie logicielle
  - 4.9.1. Concepts de base: mesures, métriques et indicateurs
  - 4.9.2. Types de métriques en génie logiciel
  - 4.9.3. Le processus de mesure
  - 4.9.4. ISO 25024. Mesures externes et de qualité utilisées
  - 4.9.5. Métriques orientées objet
- 4.10. Maintenance et réingénierie des logiciels
  - 4.10.1. Processus de maintenance
  - 4.10.2. Cadre du processus de maintenance standard. ISO/EIEC 14764
  - 4.10.3. Le modèle de processus d' réingénierie des de software
  - 4.10.4. Ingénierie inverse

## Module 5. Qualité et audit des systèmes d'information

- 5.1. Introduction aux Systèmes de Gestion de la Sécurité de l'Information
  - 5.1.1. Principes fondamentaux du SMSI
  - 5.1.2. Règles d'or du SMSI
  - 5.1.3. Rôle de l'audit informatique dans les SGSI
- 5.2. La planification dans la gestion de la sécurité
  - 5.2.1. Concepts liés à la gestion de la sécurité
  - 5.2.2. Classification de l'information: objectifs, concepts et rôles
  - 5.2.3. Mise en œuvre des politiques de sécurité: politiques, normes et procédures de sécurité
  - 5.2.4. Gestion des risques: principes et analyse des risques liés aux actifs informationnels
- 5.3. Principaux mécanismes de protection des actifs informationnels I
  - 5.3.1. Aperçu des principaux outils cryptographiques pour la protection de la triade CID
  - 5.3.2. Prise en compte de la vie privée, de l'anonymat et des exigences de gestion de la traçabilité des utilisateurs
- 5.4. Principaux mécanismes de protection des actifs informationnels II
  - 5.4.1. Sécurité des communications: protocoles, dispositifs et architectures de sécurité
  - 5.4.2. Sécurité des systèmes d'exploitation
- 5.5. Contrôle interne des SGSI
  - 5.5.1. Taxonomie des contrôles du SGSI: contrôles administratifs, logiques et physiques
  - 5.5.2. Classification des contrôles en fonction de la manière dont ils traitent la menace: contrôles pour la prévention, la détection et la remédiation des menaces
  - 5.5.3. Mise en œuvre de systèmes de contrôle interne dans les SGSI
- 5.6. Types d'audits
  - 5.6.1. Différence entre l'audit et le contrôle interne
  - 5.6.2. Audit interne et externe
  - 5.6.3. Classification de l'audit en fonction de l'objectif et du type d'analyse
- 5.7. Scénariste et scénario: sujet et objet protégés par la Propriété Intellectuelle
  - 5.7.1. Introduction aux tests de pénétration et à l'analyse médico-légale
  - 5.7.2. Définition et pertinence des concepts de *fingerprinting* et *footprinting*
- 5.8. Analyse de vulnérabilité et surveillance du trafic réseau
  - 5.8.1. Outils pour l'analyse de la vulnérabilité des systèmes
  - 5.8.2. Principales vulnérabilités dans le contexte des applications web
  - 5.8.3. Analyse des protocoles de communication
- 5.9. Le processus d'audit informatique
  - 5.9.1. Concept de cycle de vie du développement des systèmes
  - 5.9.2. Surveillance des activités et des processus: collecte et traitement des données
  - 5.9.3. Méthodologies d'audit informatique
  - 5.9.4. Le processus d'audit informatique
  - 5.9.5. Identification des principales infractions et délits dans le contexte informatique
  - 5.9.6. Enquête sur la criminalité informatique: introduction à l'analyse judiciaire et à sa relation avec l'audit informatique
- 5.10. Plans de continuité des activités et de reprise après sinistre
  - 5.10.1. Définition du plan de continuité des activités et du concept d'interruption des activités
  - 5.10.2. Recommandation du NIST sur les plans de continuité des activités
  - 5.10.3. Plan de reprise après sinistre
  - 5.10.4. Processus de plan de reprise après sinistre

**Module 6. Intégration des systèmes**

- 6.1. Introduction aux systèmes d'information d'entreprise
  - 6.1.1. Le rôle des systèmes d'information
  - 6.1.2. Conception des systèmes d'information?
  - 6.1.3. Dimensions des systèmes d'information
  - 6.1.4. Processus d'affaires et systèmes d'information
  - 6.1.5. Le département IS/IT
- 6.2. Opportunités et besoins en matière de systèmes d'information dans l'entreprise
  - 6.2.1. Organisations et systèmes d'information
  - 6.2.2. Caractéristiques des organisations
  - 6.2.3. Impact des systèmes d'information sur l'entreprise
  - 6.2.4. Les systèmes d'information pour un avantage concurrentiel
  - 6.2.5. Utilisation de systèmes dans l'administration et la gestion des affaires
- 6.3. Concepts de base des systèmes et technologies de l'information
  - 6.3.1. Données, informations et connaissances
  - 6.3.2. Technologie et systèmes d'information
  - 6.3.3. Composants technologiques
  - 6.3.4. Classification et types de systèmes d'information
  - 6.3.5. Architectures basées sur les services et les processus d'entreprise
  - 6.3.6. Formes d'intégration des systèmes
- 6.4. Systèmes intégrés de gestion des ressources de l'entreprise
  - 6.4.1. Exigences commerciales
  - 6.4.2. Un système d'information intégré pour l'entreprise
  - 6.4.3. Acquisition vs. Développement
  - 6.4.4. Mise en œuvre de l' ERP
  - 6.4.5. Implications pour la gestion
  - 6.4.6. Principaux fournisseurs d'ERP
- 6.5. Systèmes d'information pour la gestion de la chaîne d'approvisionnement et de la relation client
  - 6.5.1. Définition de chaîne d'approvisionnement
  - 6.5.2. Gestion effets de la chaîne d'approvisionnement
  - 6.5.3. Le rôle des systèmes d'information
  - 6.5.4. Solutions pour la gestion de chaîne d'approvisionnement
  - 6.5.5. La gestion des relations avec les clients
  - 6.5.6. Le rôle des systèmes d'information
  - 6.5.7. Mise en œuvre d'un système CRM
  - 6.5.8. Facteurs critiques de succès dans la mise en œuvre de la CRM
  - 6.5.9. CRM, e-CRM et autres tendances
- 6.6. Prise de décision en matière d'investissement dans les TIC et planification des systèmes d'information
  - 6.6.1. Critères de décision en matière d'investissement dans les TIC
  - 6.6.2. Lier le projet au plan de gestion et d'affaires
  - 6.6.3. Implications en termes de gestion
  - 6.6.4. Refonte des processus d'affaires
  - 6.6.5. Décision de la direction sur les méthodologies de mise en œuvre
  - 6.6.6. Nécessité d'une planification des systèmes d'information
  - 6.6.7. Objectifs, participants et calendrier
  - 6.6.8. Structure et développement du plan des systèmes
  - 6.6.9. Suivi et mise à jour
- 6.7. Considérations de sécurité dans l'utilisation des TIC
  - 6.7.1. Analyse des risques
  - 6.7.2. Sécurité des systèmes d'information
  - 6.7.3. Conseils pratiques
- 6.8. Faisabilité de la mise en œuvre de projets TIC et aspects financiers dans les projets de systèmes d'information
  - 6.8.1. Description et objectifs
  - 6.8.2. Participants au SVE
  - 6.8.3. Techniques et pratiques
  - 6.8.4. Structure des coûts
  - 6.8.5. Projection financière
  - 6.8.6. Budgets

- 6.9. *Business Intelligence*
  - 6.9.1. Qu'est-ce que la Business Intelligence?
  - 6.9.2. Stratégie et mise en œuvre de la BI
  - 6.9.3. Le présent et l'avenir de la BI
- 6.10. ISO/IEC 12207
  - 6.10.1. Qu'est-ce que "ISO/IEC 12207"?
  - 6.10.2. Analyse des Systèmes d'Informations
  - 6.10.3. Conception d'un Système d'Information
  - 6.10.4. Mise en œuvre et acceptation du Système d'Information

## Module 7. Réutilisation de software

- 7.1. Présentation de réutilisation software
  - 7.1.1. Qu'est-ce que la la réutilisation des logiciels
  - 7.1.2. Avantages et inconvénients de la réutilisation des logiciels
  - 7.1.3. Principales techniques de réutilisation des logiciels
- 7.2. Introduction aux modèles de conception
  - 7.2.1. Qu'est-ce qu'un modèle de conception?
  - 7.2.2. Catalogue des principaux modèles de conception
  - 7.2.3. Comment utiliser les modèles pour résoudre les problèmes de conception
  - 7.2.4. Comment choisir le meilleur modèle de conception
- 7.3. Modèles de création I
  - 7.3.1. Modèles de création
  - 7.3.2. Modèle de fabrique abstraite
  - 7.3.3. Exemple de mise en œuvre du modèle de fabrique abstraite
  - 7.3.4. Modèle de construction
  - 7.3.5. Exemple de mise en œuvre du constructeur
  - 7.3.6. Usine abstraite vs. Builder
- 7.4. Modèles de création II
  - 7.4.1. Patron Factory Method
  - 7.4.2. Factory Method vs. Abstract Factory
  - 7.4.3. Modèle Singleton
- 7.5. Modèles structurels I
  - 7.5.1. Modèles structurels
  - 7.5.2. Modèle Adapter
  - 7.5.3. Modèle Bridge
- 7.6. Modèle structurels II
  - 7.6.1. Modèle Composite
  - 7.6.2. Modèle Decorator
- 7.7. Modèle structurels III
  - 7.7.1. Modèle Facade
  - 7.7.2. Modèle Proxy
- 7.8. Modèle de comportement I
  - 7.8.1. Concept de modèles comportementaux
  - 7.8.2. Modèle de comportement: chaîne de responsabilité
  - 7.8.3. Modèle de comportement Ordre
- 7.9. Modèle de comportement II
  - 7.9.1. Modèle d'Interprète ou *Interpreter*
  - 7.9.2. Modèle Iterador
  - 7.9.3. Modèle Observador
  - 7.9.4. Modèle de Stratégie
- 7.10. *Frameworks*
  - 7.10.1. Concept de framework
  - 7.10.2. Développement à l'aide de frameworks
  - 7.10.3. Model View Controller
  - 7.10.4. *Framework* pour la conception d'interfaces utilisateur graphiques
  - 7.10.5. *Frameworks* pour le développement d'applications web
  - 7.10.6. *Frameworks* pour gérer la persistance des objets dans les bases de données

## Module 8. Services informatiques

- 8.1. La transformation digitale I
  - 8.1.1. L'innovation dans l'entreprise
  - 8.1.2. La gestion de la production
  - 8.1.3. La gestion financière
- 8.2. La transformation digitale II
  - 8.2.1. Le marketing
  - 8.2.2. La gestion de Ressources Humaines
  - 8.2.3. Un système d'information complet
- 8.3. Étude de cas
  - 8.3.1. Présentation de l'entreprise
  - 8.3.2. Méthodologie pour analyser l'acquisition de TI
  - 8.3.3. Détermination des coûts, bénéfices et risques
  - 8.3.4. Évaluation économique de l'investissement
- 8.4. la gouvernance et la gestion des TIC
  - 8.4.1. Définition de la gouvernance des technologies et des systèmes d'information
  - 8.4.2. Différence entre la gouvernance et la gestion de l'IST
  - 8.4.3. Cadres pour la gouvernance et la gestion des TSI
  - 8.4.4. Normes, gouvernance et gestion des TSI
- 8.5. Gouvernance d'entreprise dans le domaine des TIC
  - 8.5.1. Qu'est-ce qu'une bonne gouvernance d'entreprise?
  - 8.5.2. Contexte de la gouvernance des TIC
  - 8.5.3. Norme ISO/IEC 38500:2008
  - 8.5.4. Mise en œuvre d'une bonne gouvernance des TIC
  - 8.5.5. Gouvernance et meilleures pratiques en matière de TIC
  - 8.5.6. La gouvernance d'entreprise. Vue d'ensemble et tendances
- 8.6. Objectifs de contrôle pour les technologies de l'information et les technologies connexes (COBIT)
  - 8.6.1. Cadre de mise en œuvre
  - 8.6.2. Domaine: planification et organisation
  - 8.6.3. Domaine: approvisionnement et mise en œuvre
  - 8.6.4. Domaine: livraison et soutien
  - 8.6.5. Domaine: suivi et évaluation
  - 8.6.6. Application du Guide COBIT
- 8.7. La bibliothèque d'infrastructure des technologies de l'information (ITIL)
  - 8.7.1. Introduction à ITIL
  - 8.7.2. Stratégies de service
  - 8.7.3. Conception des services
  - 8.7.4. Transition des services
  - 8.7.5. Opération des services
  - 8.7.6. Amélioration des services
- 8.8. Système de gestion des services
  - 8.8.1. Principes de base de la norme UNE-ISO/IEC 20000-1
  - 8.8.2. La structure de la série de normes ISO/CEI 20000
  - 8.8.3. Exigences du système de gestion des services
  - 8.8.4. Conception et transition de services nouveaux ou modifiés
  - 8.8.5. Processus de prestation de services
  - 8.8.6. Groupes de processus
- 8.9. Système de gestion des actifs des Software
  - 8.9.1. Justification du besoin
  - 8.9.2. Antécédents
  - 8.9.3. Présentation de 19770
  - 8.9.4. Mise en œuvre de la gestion
- 8.10. Gestion de la continuité des activités
  - 8.10.1. Plan de continuité des activités
  - 8.10.2. Mise en place du BCM

## Module 9. Sécurité des systèmes d'information

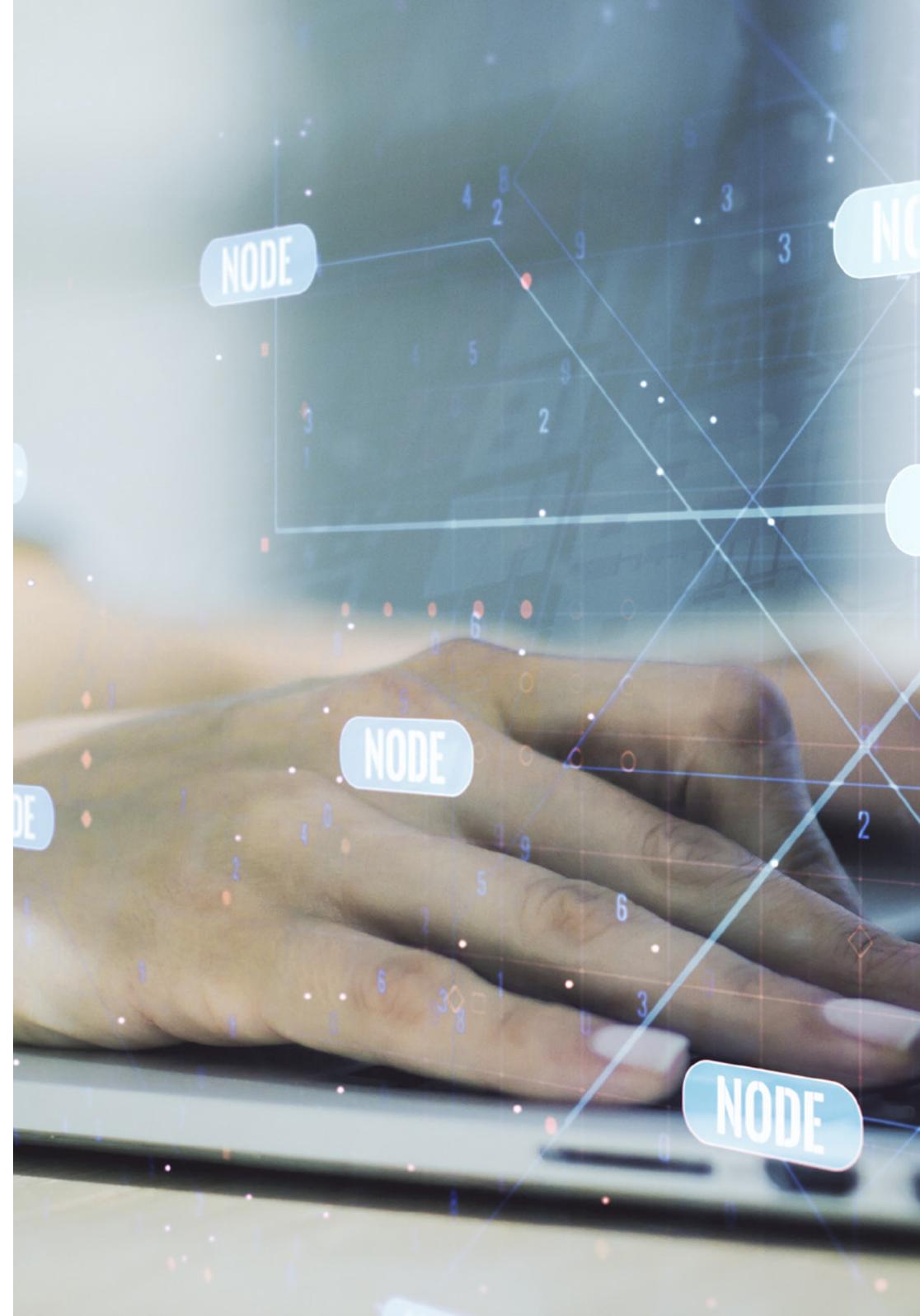
- 9.1. Un aperçu de la sécurité, de la cryptographie et des cryptanalyses classiques
  - 9.1.1. Sécurité informatique: perspective historique
  - 9.1.2. Mais qu'est-ce que la sécurité exactement?
  - 9.1.3. Histoire de la cryptographie
  - 9.1.4. Chiffres de substitution
  - 9.1.5. Étude de cas: la machine Enigma
- 9.2. Cryptographie symétrique
  - 9.2.1. Introduction et terminologie de base
  - 9.2.2. Cryptage symétrique
  - 9.2.3. Modes d'opération
  - 9.2.4. DES
  - 9.2.5. La nouvelle norme AES
  - 9.2.6. Cryptage de flux
  - 9.2.7. Cryptanalyse
- 9.3. Cryptographie asymétrique
  - 9.3.1. Origines de la cryptographie à clé publique
  - 9.3.2. Concepts de base et fonctionnement
  - 9.3.3. L'algorithme RSA
  - 9.3.4. Certificats numériques
  - 9.3.5. Stockage et gestion des clés
- 9.4. Attaques de réseau
  - 9.4.1. Menaces et attaques contre les réseaux
  - 9.4.2. Énumération
  - 9.4.3. Interception du trafic: *sniffers*
  - 9.4.4. Attaques par déni de service
  - 9.4.5. Attaques par empoisonnement ARP
- 9.5. Architectures de sécurité
  - 9.5.1. Architectures de sécurité traditionnel
  - 9.5.2. Secure Socket Layer: SSL
  - 9.5.3. Protocole SSH
  - 9.5.4. Réseaux Privés Virtuels (VPN)
  - 9.5.5. Mécanismes de protection des unités de stockage externes
  - 9.5.6. Mécanismes de protection du matériel
- 9.6. Techniques de protection des systèmes et développement de code sécurisé
  - 9.6.1. Sécurité Opérationnelle
  - 9.6.2. Ressources et contrôles
  - 9.6.3. Suivi
  - 9.6.4. Systèmes de détection des intrusions
  - 9.6.5. IDS de l'hôte
  - 9.6.6. IDS réseau
  - 9.6.7. IDS basé sur les signatures
  - 9.6.8. Systèmes de leurres
  - 9.6.9. Principes de base de la sécurité dans le développement du code
  - 9.6.10. Gestion des pannes
  - 9.6.11. Ennemi public numéro 1: les dépassements de tampon (Buffer Overflows)
  - 9.6.12. Botches cryptographiques
- 9.7. *Botnets* et spam
  - 9.7.1. Origine du problème
  - 9.7.2. Processus de spam
  - 9.7.3. Envoi de spam
  - 9.7.4. Affinement des listes de diffusion
  - 9.7.5. Techniques de protection
  - 9.7.6. Service anti- offert par des tiers
  - 9.7.7. Étude de cas
  - 9.7.8. Spam exotique
- 9.8. Audit et attaques du Web
  - 9.8.1. Collecte d'informations
  - 9.8.2. Techniques d'attaque
  - 9.8.3. Outils

- 9.9. Malware et code malveillant
  - 9.9.1. Qu'est-ce qu'un malware?
  - 9.9.2. Types de malware
  - 9.9.3. Virus
  - 9.9.4. Cryptovirus
  - 9.9.5. Vers
  - 9.9.6. Adware
  - 9.9.7. Spyware
  - 9.9.8. Canulars
  - 9.9.9. Pishing
  - 9.9.10. Trojans
  - 9.9.11. L'économie de Malware
  - 9.9.12. Solutions possibles
- 9.10. Analyse médico-légale
  - 9.10.1. Collecte des preuves
  - 9.10.2. Analyse des preuves
  - 9.10.3. Techniques anti-forensic
  - 9.10.4. Étude de cas pratique

## Module 10. Gestion de projets

- 10.1. Concepts fondamentaux de la conduite de projet et du cycle de vie de la gestion de projets
  - 10.1.1. Qu'est-ce qu'un projet?
  - 10.1.2. Méthodologie commune
  - 10.1.3. Qu'est-ce que la gestion/leadership de projet?
  - 10.1.4. Qu'est-ce qu'un plan de projet?
  - 10.1.5. Bénéfices
  - 10.1.6. Cycle de vie d'un projet
  - 10.1.7. Groupes de processus ou cycle de vie de la gestion de projet
  - 10.1.8. La relation entre les groupes de processus et les domaines de connaissances
  - 10.1.9. Relations entre le cycle de vie du produit et du projet
- 10.2. Initiation et planification
  - 10.2.1. De l'idée au projet
  - 10.2.2. Élaboration de la charte du projet
  - 10.2.3. Réunion de lancement du projet
  - 10.2.4. Tâches, connaissances et compétences dans le processus de démarrage
  - 10.2.5. Le plan de projet
  - 10.2.6. Élaboration du plan de base. Étapes
  - 10.2.7. Tâches, connaissances et compétences dans le processus de planification
- 10.3. Gestion des *stakeholders* et du champ d'application
  - 10.3.1. Identifier les parties prenantes
  - 10.3.2. Développer le plan de gestion des parties prenantes
  - 10.3.3. Gérer l'engagement des parties prenantes
  - 10.3.4. Suivre l'engagement des parties prenantes
  - 10.3.5. Objectif du projet
  - 10.3.6. La gestion de la portée et son plan
  - 10.3.7. Recueillir les besoins
  - 10.3.8. Définir l'énoncé de la portée
  - 10.3.9. Créer l'OTP
  - 10.3.10. Vérifier et contrôle de la portée
- 10.4. Élaboration du calendrier
  - 10.4.1. La gestion du temps et son plan
  - 10.4.2. Définir les activités
  - 10.4.3. Établir la séquence des activités
  - 10.4.4. Estimation des ressources pour les activités
  - 10.4.5. Estimation de la durée des activités
  - 10.4.6. Élaboration du calendrier et calcul du chemin critique
  - 10.4.7. Contrôle des horaires

- 10.5. Élaboration du budget et réponse aux risques
  - 10.5.1. Estimation des coûts
  - 10.5.2. Élaboration du budget et de la courbe en S
  - 10.5.3. Contrôle des coûts et méthode de la valeur acquise
  - 10.5.4. Concepts de risque
  - 10.5.5. Comment faire une analyse de risque
  - 10.5.6. L'élaboration du plan d'intervention
- 10.6. Gestion de la qualité
  - 10.6.1. Planification de la qualité
  - 10.6.2. Assurance de la qualité
  - 10.6.3. Contrôle de la qualité
  - 10.6.4. Concepts statistiques fondamentaux
  - 10.6.5. Outils de la gestion de la qualité
- 10.7. Communication et ressources humaines
  - 10.7.1. Planification de la gestion des communications
  - 10.7.2. Analyse des besoins en communication
  - 10.7.3. Technologie des communications
  - 10.7.4. Modèles de communication
  - 10.7.5. Méthodes de communication
  - 10.7.6. Plan de gestion de la communication
  - 10.7.7. Gestion des communications
  - 10.7.8. La gestion des ressources humaines
  - 10.7.9. Principaux acteurs et leurs rôles dans les projets
  - 10.7.10. Types d'organisations
  - 10.7.11. Organisation par projet
  - 10.7.12. L'équipe de travail



- 10.8. Approvisionnement
  - 10.8.1. Le processus de passation de marchés
  - 10.8.2. Plan
  - 10.8.3. Recherche de fournisseurs et appels d'offres
  - 10.8.4. Attribution du contrat
  - 10.8.5. Administration du contrat
  - 10.8.6. Contrats
  - 10.8.7. Types de contrats
  - 10.8.8. Négociation de contrats
- 10.9. Exécution, suivi et contrôle et clôture
  - 10.9.1. Les groupes de processus
  - 10.9.2. Mise en œuvre du projet
  - 10.9.3. Suivi et contrôle des projets
  - 10.9.4. Clôture du projet
- 10.10. Responsabilité professionnelle
  - 10.10.1. Responsabilité professionnelle
  - 10.10.2. Caractéristiques de la responsabilité sociale et professionnelle
  - 10.10.3. Code d'éthique du chef de projet
  - 10.10.4. Responsabilité vs. PMP®
  - 10.10.5. Exemples de responsabilité
  - 10.10.6. Avantages de la professionnalisation

# 06

## Stage Pratique

Une fois la période théorique en ligne terminée, ce Mastère Hybride en Ingénierie Avancée du Software comprend une Formation Pratique au sein d'une entité de référence. Pendant cette période, les diplômés bénéficieront du soutien d'un tuteur qui les aidera à tirer le meilleur parti de cette expérience. Grâce à cela, les informaticiens acquerront des compétences avancées pour connaître un saut qualitatif important dans leur carrière.



“

*Vous effectuerez votre séjour pratique dans une organisation de premier plan, où vous bénéficierez du soutien de professionnels renommés dans le domaine de l'Ingénierie Avancée du Software”*

La Formation Pratique de ce programme en Ingénierie Avancée du Software consiste en un séjour pratique dans une entité prestigieuse, d'une durée de 3 semaines, du lundi au vendredi, avec 8 heures consécutives d'enseignement pratique aux côtés d'un spécialiste associé.

De même, dans cette proposition de formation, de nature totalement pratique, les activités visent à développer et à perfectionner les compétences nécessaires pour offrir des services informatiques d'Ingénierie Avancée du Software, ainsi que les conditions qui exigent un haut niveau de qualification, orientées vers une formation spécifique pour l'exercice de l'activité.

Il s'agit d'une excellente occasion pour les informaticiens de s'immerger dans un environnement de travail réel, où ils feront partie d'une équipe de professionnels spécialisés dans ce domaine. Ainsi, les diplômés participeront à des projets liés à des sujets tels que la sécurité des systèmes d'information, la réutilisation du Software ou la conception d'applications web, entre autres.

L'enseignement pratique sera dispensé avec la participation active de l'étudiant, qui réalisera les activités et les procédures de chaque domaine de compétence (apprendre à apprendre et apprendre à faire), avec l'accompagnement et les conseils des enseignants et d'autres collègues formateurs qui facilitent le travail en équipe et l'intégration multidisciplinaire en tant que compétences transversales pour la pratique de l'Informatique (apprendre à être et apprendre à être en relation avec les autres).



Les procédures décrites ci-dessous constitueront la base de la partie pratique de la formation et leur mise en œuvre dépendront de la disponibilité et de la charge de travail du centre, les activités proposées étant les suivantes:

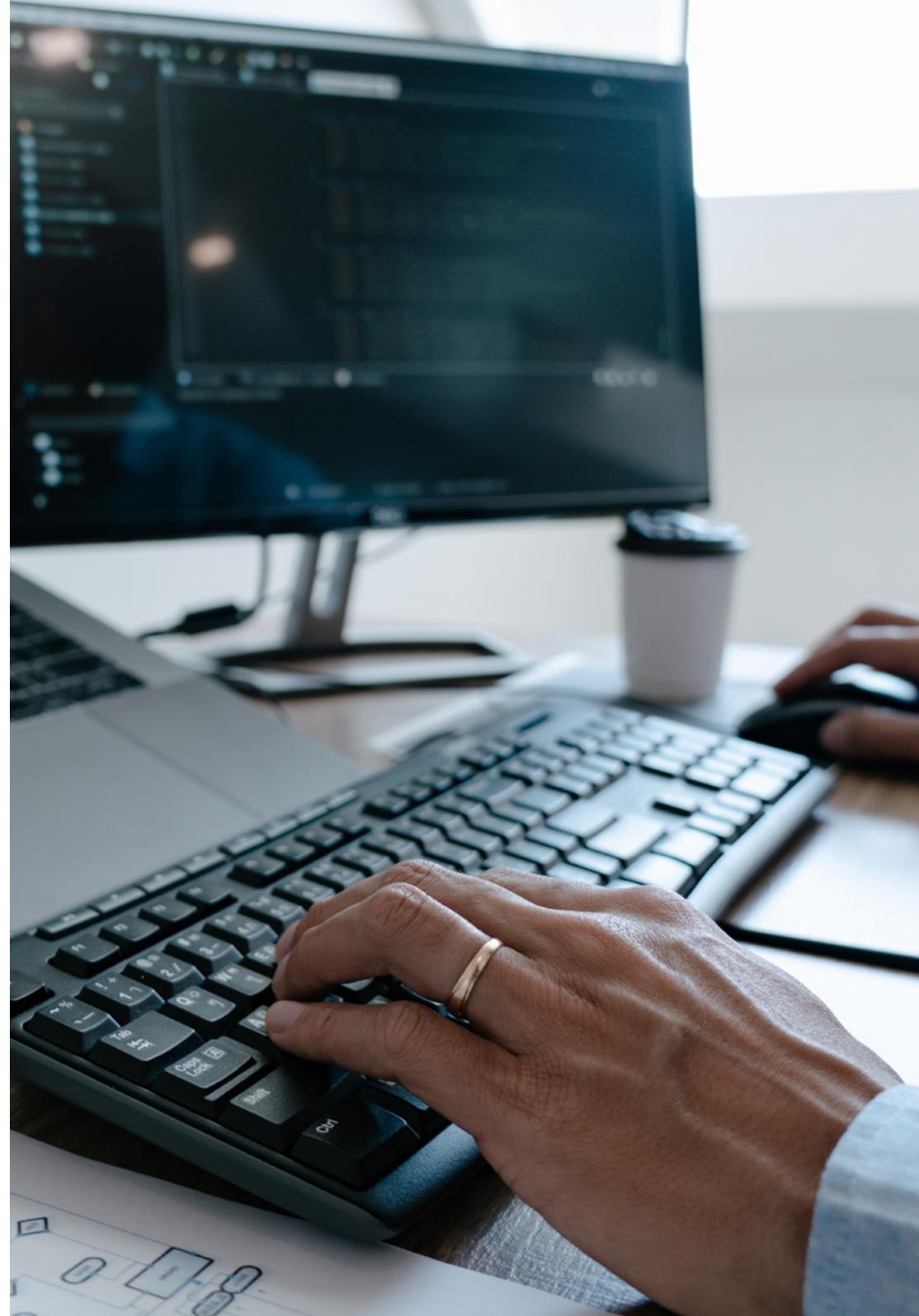
Module	Activité pratique
Développement Avancé de Software	Concevoir des architectures de software évolutives, robustes et faciles à maintenir
	Utiliser des techniques de modélisation telles que <i>Unified Modeling Language</i> pour représenter à la fois la structure et le comportement du système avant la mise en œuvre
	Écrire un code efficace et propre dans une variété de langages de programmation
	Utiliser des méthodologies agiles pour planifier, exécuter et suivre les projets de software
Gestion des Exigences	Analyser l'environnement de l'utilisateur et étudier le domaine d'application pour identifier les problèmes que le logiciel doit résoudre
	Utiliser des techniques de modélisation (telles que les diagrammes UML ou les cas d'utilisation et les scénarios) pour représenter les exigences de manière structurée
	Rédiger les documents de spécification des exigences avec les <i>stakeholders</i> pour s'assurer que les exigences saisies sont correctes
	Développer des prototypes et des simulations du système afin de valider les exigences avec les utilisateurs
Gestion des Exigences	Établir des normes de qualité pour la maintenance des systèmes d'information, sur la base de <i>frameworks</i>
	Effectuer des tests fonctionnels, de performance, de sécurité et d'utilisabilité
	Identifier et évaluer les risques associés aux systèmes d'information
	Mettre en œuvre des processus d'intégration et de déploiement continus afin de garantir que les nouvelles versions du software sont développées, testées et déployées
Connexion des Systèmes	Se plonger dans les systèmes pour identifier les défis potentiels de l'intégration et planifier des solutions efficaces
	Créer des modèles de données qui définissent la manière dont les données seront structurées et traitées entre les systèmes intégrés
	Concevoir l'architecture d'intégration, en sélectionnant les modèles et les méthodes les plus appropriés (par exemple, l'intégration basée sur les services, la messagerie, les API)
	Configurer les interfaces pour permettre la communication et le transfert de données entre les systèmes intégrés

## Assurance responsabilité civile

La principale préoccupation de cette institution est de garantir la sécurité des stagiaires et des autres collaborateurs nécessaires aux processus de formation pratique dans l'entreprise. Parmi les mesures destinées à atteindre cet objectif figure la réponse à tout incident pouvant survenir au cours de la formation d'apprentissage.

Pour ce faire, cette université s'engage à souscrire une police d'assurance responsabilité civile pour couvrir toute éventualité pouvant survenir pendant le séjour au centre de stage.

Cette police d'assurance couvrant la responsabilité civile des stagiaires doit être complète et doit être souscrite avant le début de la période de Formation Pratique. Ainsi, le professionnel n'a pas à se préoccuper des imprévus et bénéficiera d'une couverture jusqu'à la fin du stage pratique dans le centre.



## Conditions générales de la formation pratique

Les conditions générales de la Convention de Stage pour le programme sont les suivantes:

**1. TUTEUR:** Pendant le Mastère Hybride, l'étudiant se verra attribuer deux tuteurs qui l'accompagneront tout au long du processus, en résolvant tous les doutes et toutes les questions qui peuvent se poser. D'une part, il y aura un tuteur professionnel appartenant au centre de placement qui aura pour mission de guider et de soutenir l'étudiant à tout moment. D'autre part, un tuteur académique sera également assigné à l'étudiant, et aura pour mission de coordonner et d'aider l'étudiant tout au long du processus, en résolvant ses doutes et en lui facilitant tout ce dont il peut avoir besoin. De cette manière, le professionnel sera accompagné à tout moment et pourra consulter les doutes qui pourraient surgir, tant sur le plan pratique que sur le plan académique.

**2. DURÉE:** Le programme de formation pratique se déroulera sur trois semaines continues, réparties en journées de 8 heures, cinq jours par semaine. Les jours de présence et l'emploi du temps relèvent de la responsabilité du centre, qui en informe dûment et préalablement le professionnel, et suffisamment à l'avance pour faciliter son organisation.

**3. ABSENCE:** En cas de non présentation à la date de début du Mastère Hybride, l'étudiant perdra le droit au stage sans possibilité de remboursement ou de changement de dates. Une absence de plus de deux jours au stage, sans raison médicale justifiée, entraînera l'annulation du stage et, par conséquent, la résiliation automatique du contrat. Tout problème survenant au cours du séjour doit être signalé d'urgence au tuteur académique.

**4. CERTIFICATION:** Les étudiants qui achèvent avec succès le Mastère Hybride recevront un certificat accréditant le séjour pratique dans le centre en question.

**5. RELATION DE TRAVAIL:** Le Mastère Hybride ne constituera en aucun cas une relation de travail de quelque nature que ce soit.

**6. PRÉREQUIS:** Certains centres peuvent être amenés à exiger des références académiques pour suivre le Mastère Hybride. Dans ce cas, il sera nécessaire de le présenter au département de formations de TECH afin de confirmer l'affectation du centre choisi.

**7. NON INCLUS:** Le mastère Hybride n'inclut aucun autre élément non mentionné dans les présentes conditions. Par conséquent, il ne comprend pas l'hébergement, le transport vers la ville où le stage a lieu, les visas ou tout autre avantage non décrit.

Toutefois, les étudiants peuvent consulter leur tuteur académique en cas de doutes ou de recommandations à cet égard. Ce dernier lui fournira toutes les informations nécessaires pour faciliter les démarches.

# 07

## Où puis-je effectuer mon Stage Pratique?

Fidèle à sa philosophie de fournir des diplômes universitaires de haute qualité, TECH élargit les opportunités académiques pour les étudiants et leur permet d'effectuer leur formation pratique dans différentes entités de prestige international. Les informaticiens ont ainsi une occasion idéale d'améliorer leur qualité professionnelle en travaillant avec les meilleurs spécialistes dans le domaine de l'Ingénierie Avancée du Software.





“

*Vous effectuerez votre séjour pratique dans une institution de référence en Ingénierie Avancée du Software”*

## tech 46 | Où puis-je effectuer mon Stage Pratique?



Les étudiants peuvent suivre la partie pratique de ce Mastère Hybride dans les centres suivants:



**Informatique**

### Captia Ingeniería

Pays	Ville
Espagne	Madrid

Adresse: Av. de las Nieves, 37, Bloque A Planta 1  
Oficina E, 28935, Móstoles, Madrid

Entreprise informatique qui se consacre à la fourniture de solutions technologiques avancées aux industries

---

**Formations pratiques connexes:**

- Visual Analytics et Big Data
- Développement de Software





“

*Comprenez mieux la théorie la plus pertinente dans ce domaine, puis appliquez-la dans un environnement de travail réel”*

08

# 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

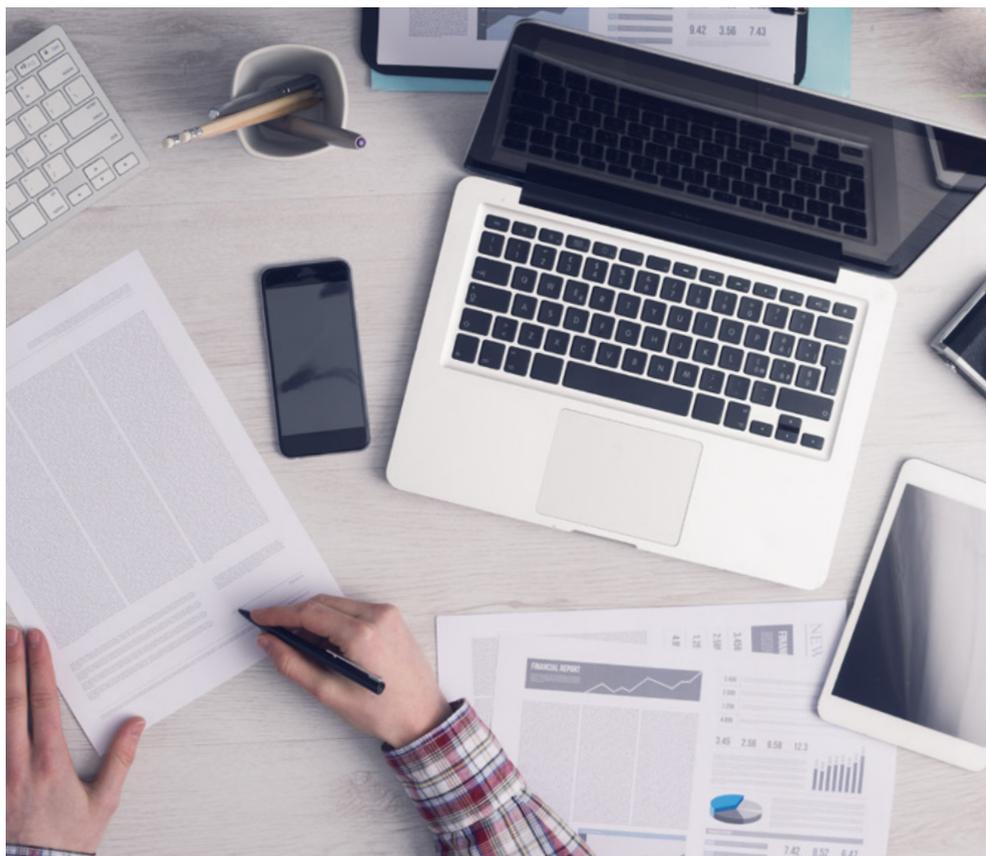
Notre programme offre une méthode révolutionnaire de développement des compétences et des connaissances. Notre objectif est de renforcer les compétences dans un contexte changeant, compétitif et hautement exigeant.

“

*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.

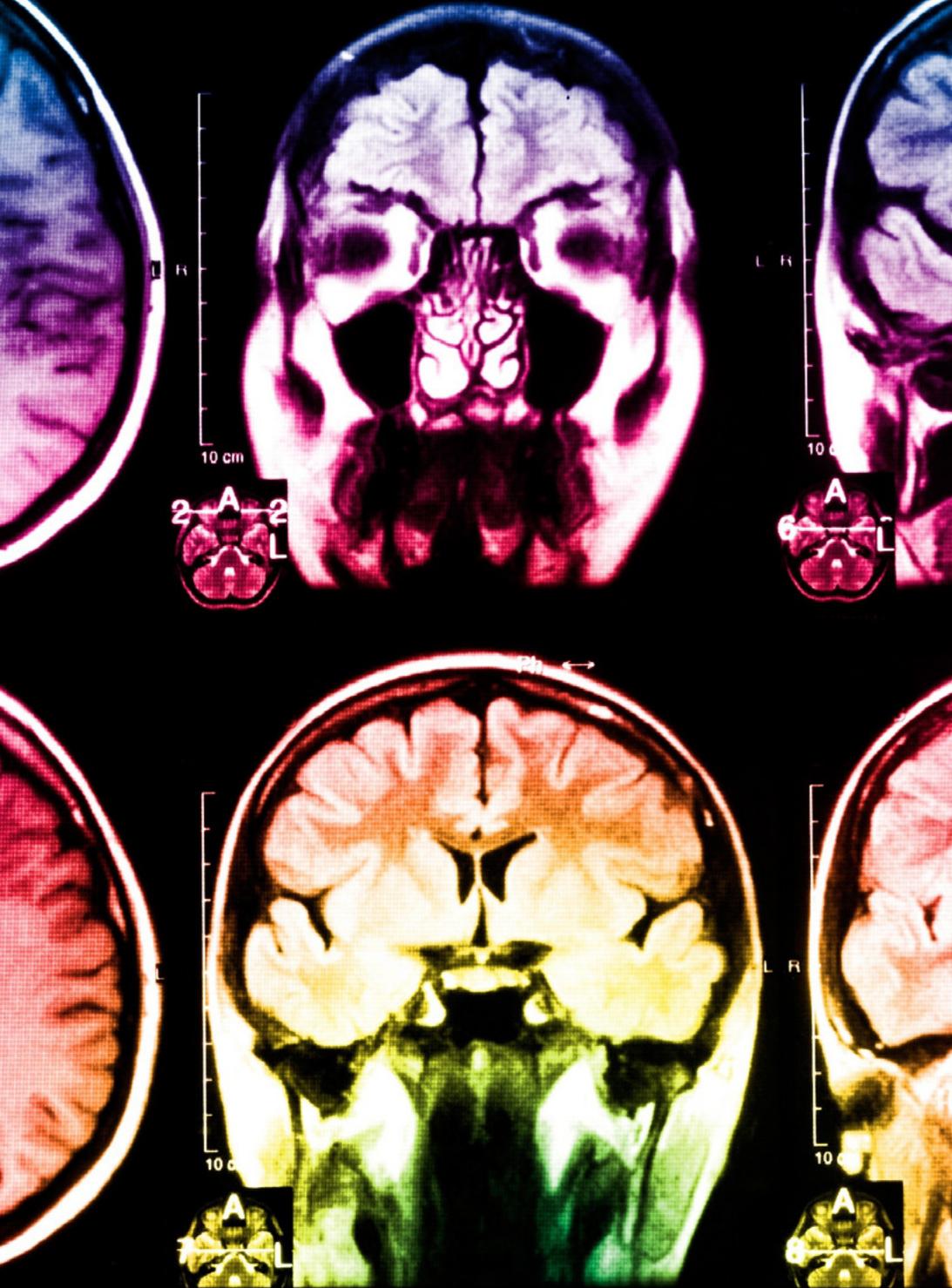
Nous enrichissons l'Étude de Cas avec la meilleure méthode d'enseignement 100% en ligne: le Relearning.

*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.



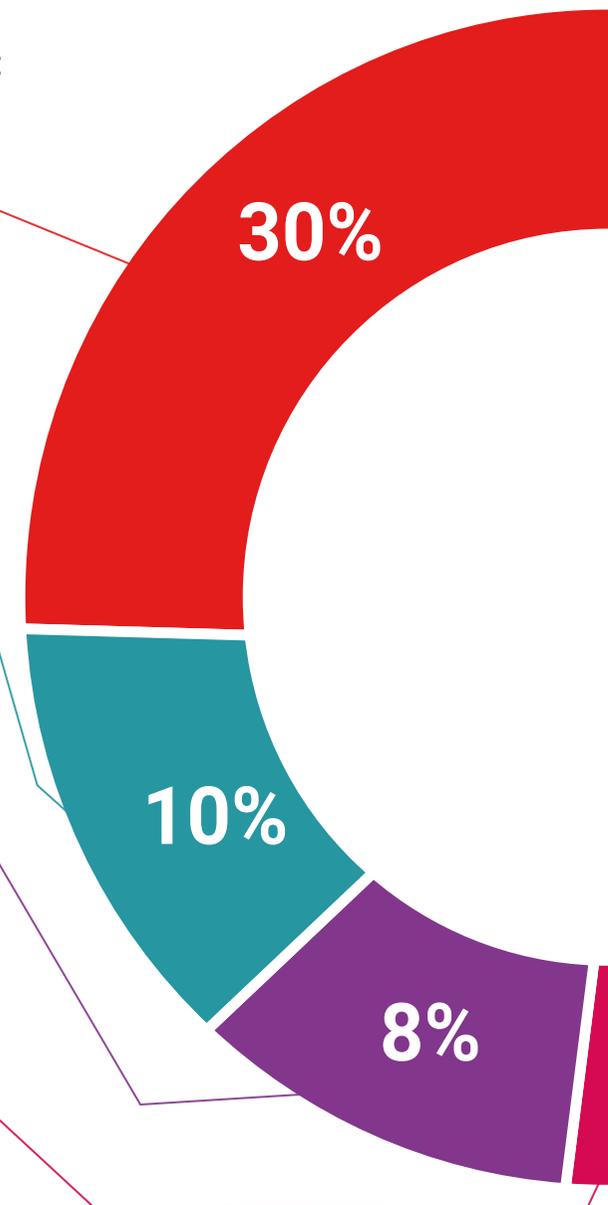
#### Pratiques en compétences et aptitudes

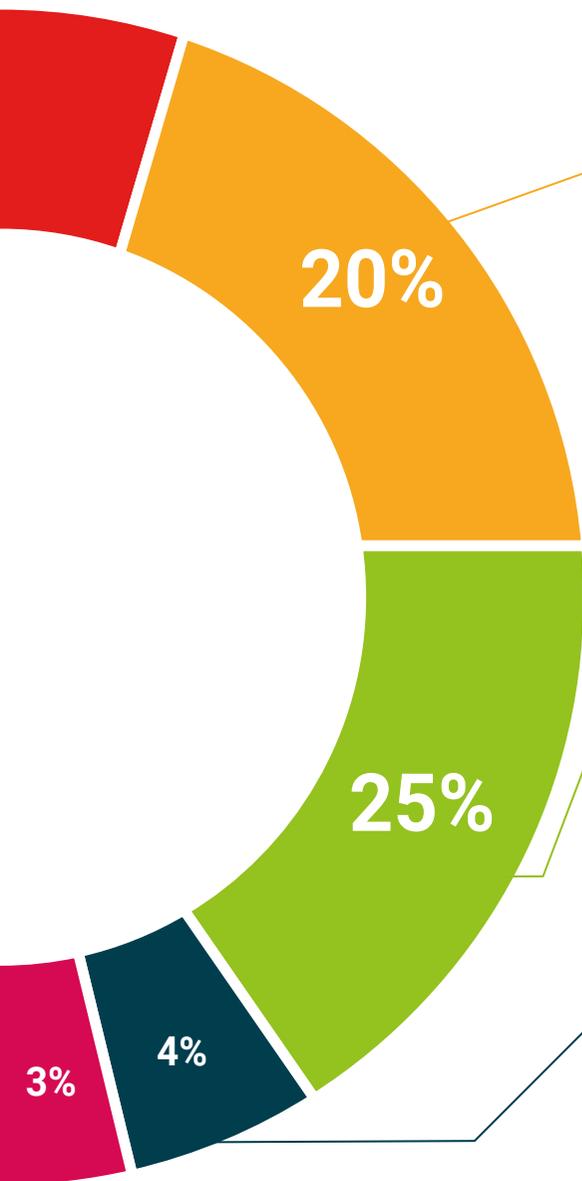
Les étudiants réaliseront des activités visant à développer des compétences et des aptitudes spécifiques dans chaque domaine. Des activités pratiques et dynamiques pour acquérir et développer les compétences et aptitudes qu'un spécialiste doit développer dans le cadre de la mondialisation dans laquelle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



# 09 Diplôme

Le diplôme de Mastère Hybride en Ingénierie Avancée du Software garantit, en plus de la formation la plus rigoureuse et actualisée, l'accès à un diplôme de Mastère Hybride délivré par TECH Université Technologique.



“

*Terminez ce programme avec succès et obtenez votre diplôme universitaire sans avoir à vous déplacer ou à passer par des procédures fastidieuses”*

Ce diplôme de **Mastère Hybride en Ingénierie Avancée du Software** contient le programme le plus complet et le plus actuel sur la scène professionnelle et académique.

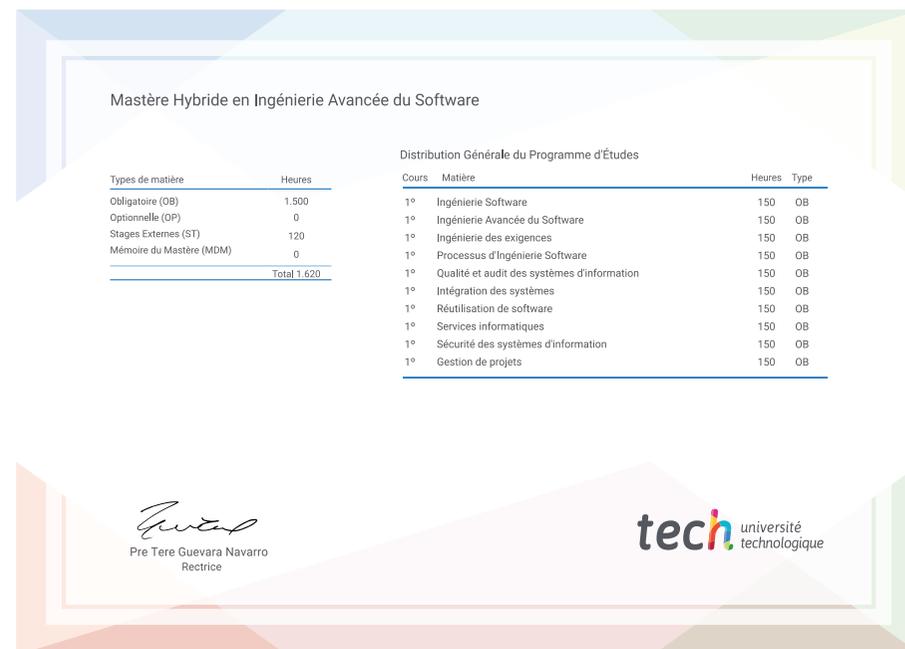
Une fois que l'étudiant aura réussi les évaluations, il recevra par courrier, avec accusé de réception, le diplôme de Mastère Hybride correspondant délivré par TECH.

En plus du Diplôme, vous pourrez obtenir un certificat, ainsi qu'une attestation du contenu du programme. Pour ce faire, vous devez contacter votre conseiller académique, qui vous fournira toutes les informations nécessaires.

Diplôme: **Mastère Hybride en Ingénierie Avancée du Software**

Modalité: **Hybride (en ligne + Stage Pratique)**

Durée: **12 mois**



\*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 **tech** université  
technologique

en ligne formation

développement institutions  
Mastère Hybride  
Ingénierie Avancée du Software

classe virtuelle langage  
Modalité: Hybride (en ligne + Stage Pratique)  
Durée: 12 mois  
Diplôme: TECH Université Technologique

# Mastère Hybride

## Ingénierie Avancée du Software

