

Mastère Avancé Ingénierie du Logiciel





Mastère Avancé Ingénierie du Logiciel

- » Modalité: en ligne
- » Durée: 2 ans
- » Qualification: TECH Université Technologique
- » Horaire: à votre rythme
- » Examens: en ligne

Accès au site web: www.techtitute.com/fr/informatique/mastere-avance/mastere-avance-ingenierie-logiciel

Sommaire

01

Présentation du programme

page 4

02

Pourquoi étudier à TECH?

page 8

03

Programme d'études

page 12

04

Objectifs pédagogiques

page 36

05

Opportunités de carrière

page 42

06

Méthodologie d'étude

page 46

07

Corps Enseignant

page 56

08

Diplôme

page 60

01

Présentation du programme

La demande de Software a connu une croissance exponentielle au cours des dernières années. L'émergence de nouvelles plateformes numériques, de matériel avancé et la virtualisation croissante des processus quotidiens ont posé des défis constants aux ingénieurs en Software. Face à un public de plus en plus habitué aux technologies émergentes et dont les attentes sont de plus en plus élevées, les développeurs doivent s'adapter et créer des solutions qui répondent aux exigences du marché. Cela exige des compétences techniques dans de multiples domaines de la connaissance informatique. C'est pourquoi TECH a conçu ce programme, qui aborde les compétences essentielles pour relever les défis d'aujourd'hui, notamment la planification et la conception de Logiciels, la mise en œuvre de mesures de sécurité, les méthodes de réutilisation des Logiciels et l'administration des serveurs web.



“

*Jouez un rôle clé dans l'avenir technologique.
Spécialisez-vous dans l'Ingénierie du Logiciel
et créez des systèmes transformateurs”*

Le Génie Logiciel se positionne comme l'une des disciplines les plus pertinentes dans l'ère technologique actuelle, car pratiquement tout le monde interagit quotidiennement avec des appareils numériques, ce qui a un impact significatif sur les économies mondiales. Dans ce contexte, le rôle des ingénieurs en logiciel acquiert une grande importance, car la combinaison de connaissances techniques et de méthodologies innovantes permet de créer des solutions efficaces adaptées aux besoins du marché. Une spécialisation poussée dans ce domaine est essentielle pour garantir le succès des projets technologiques. C'est pourquoi TECH a conçu ce programme académique complet, destiné à offrir aux professionnels les outils les plus avancés et les plus récents.

Dans cette optique, le programme aborde les concepts clés du Génie Logiciel, en approfondissant des aspects fondamentaux tels que le développement d'applications, les plates-formes technologiques et les méthodologies les plus avancées pour la gestion de projets. La partie technique joue un rôle important, guidant l'étudiant de la conception des idées à leur mise en œuvre dans des systèmes fonctionnels. Une occasion unique de comprendre l'architecture Logicielle et la création de solutions innovantes, tout en explorant les dernières tendances en matière de programmation, de développement mobile et d'applications en nuage. En outre, pour renforcer les compétences en leadership, le contenu inclut les concepts les plus récents en matière de gestion de projet et de méthodologies agiles, une valeur ajoutée pour les professionnels expérimentés qui cherchent à mettre à jour leurs compétences ou pour ceux qui veulent atteindre des postes stratégiques pour la première fois.

L'un des principaux avantages de ce programme est son mode 100% en ligne, qui élimine la nécessité de voyager ou de s'adapter à des horaires rigides. Cela permet aux étudiants de gérer leur apprentissage de manière autonome, en organisant leur temps et leur rythme d'étude en fonction de leurs propres besoins, ce qui est idéal pour le combiner avec d'autres responsabilités quotidiennes.

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

- ♦ Le développement d'études de cas présentées par des experts en Ingénierie du Logiciel
- ♦ Le contenu graphique, schématique et éminemment pratique du programme fournit des informations scientifiques et pratiques sur les disciplines essentielles à la pratique professionnelle
- ♦ Exercices pratiques permettant de réaliser le processus d'auto-évaluation afin d'améliorer l'apprentissage
- ♦ L'accent est mis sur les méthodologies innovantes en matière de Gestion de l'Ingénierie du Logiciel
- ♦ Cours théoriques, questions à l'expert, forums de discussion sur des sujets controversés et travail de réflexion individuel
- ♦ La possibilité d'accéder aux contenus depuis n'importe quel appareil fixe ou portable doté d'une connexion internet



Stimulez le développement technologique grâce à une gestion efficace dans le domaine de l'Ingénierie du Logiciel, la clé de la réussite organisationnelle”

“

Étudiez à votre rythme grâce à un programme 100% en ligne, accessible à tout moment et depuis n'importe quel endroit du monde”

Son corps enseignant comprend des professionnels appartenant au domaine de l'Ingénierie du Logiciel, qui apportent l'expérience de leur travail à ce programme, ainsi que des spécialistes reconnus issus de grandes entreprises 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.

Accédez à la méthodologie la plus innovante et la plus actuelle pour apprendre le génie logiciel, conçue par TECH.

Consolidez vos connaissances techniques grâce à une grande variété de ressources pratiques incluses dans ce programme.



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.



“

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

La meilleure université en ligne du monde, selon FORBES

Le prestigieux magazine Forbes, spécialisé dans les affaires et la finance, a désigné TECH comme « la meilleure université en ligne du monde ». C'est ce qu'ils ont récemment déclaré dans un article de leur édition numérique dans lequel ils se font l'écho de la réussite de cette institution, « grâce à l'offre académique qu'elle propose, à la sélection de son corps enseignant et à une méthode d'apprentissage innovante visant à former les professionnels du futur ».

Forbes
Mejor universidad
online del mundo

Plan
de estudios
más completo

Les programmes d'études les plus complets sur la scène universitaire

TECH offre les programmes d'études les plus complets sur la scène universitaire, avec des programmes qui couvrent les concepts fondamentaux et, en même temps, les principales avancées scientifiques dans leurs domaines scientifiques spécifiques. En outre, ces programmes sont continuellement mis à jour afin de garantir que les étudiants sont à la pointe du monde universitaire et qu'ils possèdent les compétences professionnelles les plus recherchées. De cette manière, les diplômés de l'université offrent à ses diplômés un avantage significatif pour propulser leur carrière vers le succès.

Le meilleur personnel enseignant top international

Le corps enseignant de TECH se compose de plus de 6 000 professeurs jouissant du plus grand prestige international. Des professeurs, des chercheurs et des hauts responsables de multinationales, parmi lesquels figurent Isaiah Covington, entraîneur des Boston Celtics, Magda Romanska, chercheuse principale au Harvard MetaLAB, Ignacio Wistumba, président du département de pathologie moléculaire translationnelle au MD Anderson Cancer Center, et D.W. Pine, directeur de la création du magazine TIME, entre autres.

Profesorado
TOP
Internacional

Une méthode d'apprentissage unique

TECH est la première université à utiliser *Relearning* dans tous ses formations. Il s'agit de la meilleure méthodologie d'apprentissage en ligne, accréditée par des certifications internationales de qualité de l'enseignement, fournies par des agences éducatives prestigieuses. En outre, ce modèle académique perturbateur est complété par la « Méthode des Cas », configurant ainsi une stratégie d'enseignement en ligne unique. Des ressources pédagogiques innovantes sont également mises en œuvre, notamment des vidéos détaillées, des infographies et des résumés interactifs.

La metodología
más eficaz

La plus grande université numérique du monde

TECH est la plus grande université numérique du monde. Nous sommes la plus grande institution éducative, avec le meilleur et le plus vaste catalogue éducatif numérique, cent pour cent en ligne et couvrant la grande majorité des domaines de la connaissance. Nous proposons le plus grand nombre de diplômes propres, de diplômes officiels de troisième cycle et de premier cycle au monde. Au total, plus de 14 000 diplômes universitaires, dans onze langues différentes, font de nous la plus grande institution éducative au monde.

n°1
Mundial
Mayor universidad
online del mundo

L'université en ligne officielle de la NBA

TECH est l'université en ligne officielle de la NBA. Grâce à un accord avec la grande ligue de basket-ball, elle offre à ses étudiants des programmes universitaires exclusifs ainsi qu'un large éventail de ressources pédagogiques axées sur les activités de la ligue et d'autres domaines de l'industrie du sport. Chaque programme est conçu de manière unique et comprend des conférenciers exceptionnels: des professionnels ayant un passé sportif distingué qui apporteront leur expertise sur les sujets les plus pertinents.

Leaders en matière d'employabilité

TECH a réussi à devenir l'université leader en matière d'employabilité. 99% de ses étudiants obtiennent un emploi dans le domaine qu'ils ont étudié dans l'année qui suit la fin de l'un des programmes de l'université. Un nombre similaire parvient à améliorer immédiatement sa carrière. Tout cela grâce à une méthodologie d'étude qui fonde son efficacité sur l'acquisition de compétences pratiques, absolument nécessaires au développement professionnel.



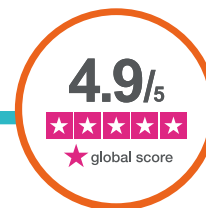
Google Partner Premier

Le géant américain de la technologie a décerné à TECH le badge Google Partner Premier. Ce prix, qui n'est décerné qu'à 3% des entreprises dans le monde, souligne l'expérience efficace, flexible et adaptée que cette université offre aux étudiants. Cette reconnaissance atteste non seulement de la rigueur, de la performance et de l'investissement maximaux dans les infrastructures numériques de TECH, mais positionne également TECH comme l'une des principales entreprises technologiques au monde.



L'université la mieux évaluée par ses étudiants

Les étudiants ont positionné TECH comme l'université la mieux évaluée du monde dans les principaux portails d'opinion, soulignant sa note la plus élevée de 4,9 sur 5, obtenue à partir de plus de 1 000 évaluations. Ces résultats consolident TECH en tant qu'institution universitaire de référence internationale, reflétant l'excellence et l'impact positif de son modèle éducatif.



03

Programme d'études

Le programme d'études de ce programme en Ingénierie du Logiciel a été soigneusement conçu par une équipe d'experts en informatique et en développement technologique. En outre, tout au long du programme, les aspects fondamentaux de l'Ingénierie du Logiciel sont abordés de manière exhaustive, couvrant tout depuis la planification, la conception et le développement d'applications jusqu'à la sécurité informatique avancée et l'administration efficace des serveurs. Les étudiants acquerront des compétences techniques spécialisées et apprendront à diriger des projets complexes qui répondent aux exigences d'un marché en constante évolution.





“

*Participez au progrès technologique
en développant des solutions
innovantes et efficaces dans le
domaine de l'Ingénierie du Logiciel”*

Module 1. Méthodologies, développement et qualité du Génie Logiciel

- 1.1. Introduction au Génie Logiciel
 - 1.1.1. Introduction
 - 1.1.2. La crise du Software
 - 1.1.3. Différences entre le Génie Logiciel et l'informatique
 - 1.1.4. Éthique et responsabilité professionnelle pour le matière de Génie Logiciel
 - 1.1.5. Fabriques de Software
- 1.2. Le processus de développement de Logiciels
 - 1.2.1. Définition
 - 1.2.2. Modèle de processus Logiciel
 - 1.2.3. Le processus unifiées de développement de Logiciels
- 1.3. Développement de Software orienté objet
 - 1.3.1. Introduction
 - 1.3.2. Principes de l'orientation objet
 - 1.3.3. Définition d'un objet
 - 1.3.4. Définition des classes
 - 1.3.5. Analyse orientée objet vs. conception orientée objet
- 1.4. Développement de Software basé sur des modèles
 - 1.4.1. La nécessité de la modélisation
 - 1.4.2. Modélisation des systèmes Software
 - 1.4.3. Modélisation des objets
 - 1.4.4. UML
 - 1.4.5. Outils CASE
- 1.5. Modélisation des applications et patrons de conception avec UML
 - 1.5.1. Modélisation avancée des exigences
 - 1.5.2. Modélisation statique avancée
 - 1.5.3. Modélisation dynamique avancée
 - 1.5.4. Modélisation des composants
 - 1.5.5. Introduction aux patrons de conception avec UML
 - 1.5.6. Adapter
 - 1.5.7. Factory
 - 1.5.8. Singleton
 - 1.5.9. Strategy
 - 1.5.10. Composite
 - 1.5.11. Façade
 - 1.5.12. Observer
- 1.6. Ingénierie dirigée par les modèles
 - 1.6.1. Introduction
 - 1.6.2. Métamodélisation des systèmes
 - 1.6.3. MDA
 - 1.6.4. DSL
 - 1.6.5. Raffinements de modèles avec OCL
 - 1.6.6. Transformations de modèles
- 1.7. Ontologies en Ingénierie Software
 - 1.7.1. Introduction
 - 1.7.2. Ingénierie ontologique
 - 1.7.3. Application des ontologies en génie Logiciel
- 1.8. Méthodologies agiles pour le développement Logiciel, Scrum
 - 1.8.1. Qu'est-ce que l'agilité dans le Logiciel?
 - 1.8.2. Le manifeste Agile
 - 1.8.3. La feuille de route d'un projet agile
 - 1.8.4. Le Product Owner
 - 1.8.5. Histoires d'utilisateur
 - 1.8.6. Planification et estimation agiles
 - 1.8.7. Mesures dans le développement agile
 - 1.8.8. Introduction à Scrum
 - 1.8.9. Rôles
 - 1.8.10. Product Backlog
 - 1.8.11. Le sprint
 - 1.8.12. Réunions
- 1.9. La méthodologie de développement de Software Lean
 - 1.9.1. Introduction
 - 1.9.2. Kanban

- 1.10. Qualité et amélioration des processus Logiciels
 - 1.10.1. Introduction
 - 1.10.2. Mesure du Software
 - 1.10.3. Tests de Logiciels
 - 1.10.4. Modèle de qualité de processus Logiciel: CMMI

Module 2. Gestion de projets de Software

- 2.1. Concepts fondamentaux de la conduite de projet et du cycle de vie de la gestion de projets
 - 2.1.1. Qu'est-ce qu'un projet?
 - 2.1.2. Méthodologie commune
 - 2.1.3. Qu'est-ce que la gestion/leadership de projet?
 - 2.1.4. Qu'est-ce qu'un plan de projet?
 - 2.1.5. Bénéfices
 - 2.1.6. Cycle de vie d'un projet
 - 2.1.7. Groupes de processus ou cycle de vie de la gestion de projet
 - 2.1.8. La relation entre les groupes de processus et les domaines de connaissances
 - 2.1.9. Relations entre le cycle de vie du produit et du projet
- 2.2. Initiation et planification
 - 2.2.1. De l'idée au projet
 - 2.2.2. Élaboration de la charte du projet
 - 2.2.3. Réunion de lancement du projet
 - 2.2.4. Tâches, connaissances et compétences dans le processus de démarrage
 - 2.2.5. Le plan de projet
 - 2.2.6. Élaboration du plan de base. Étapes
 - 2.2.7. Tâches, connaissances et compétences dans le processus de planification
- 2.3. Gestion des stakeholders et du champ d'application
 - 2.3.1. Identifier les parties prenantes
 - 2.3.2. Développer le plan de gestion des parties prenantes
 - 2.3.3. Gérer l'engagement des parties prenantes
 - 2.3.4. Suivre l'engagement des parties prenantes
 - 2.3.5. Objectif du projet
 - 2.3.6. La gestion de la portée et son plan
- 2.3.7. Recueillir les besoins
- 2.3.8. Définir l'énoncé de la portée
- 2.3.9. Créer l'OTP
- 2.3.10. Vérifier et contrôle de la portée
- 2.4. Élaboration du calendrier
 - 2.4.1. La gestion du temps et son plan
 - 2.4.2. Définir les activités
 - 2.4.3. Établir la séquence des activités
 - 2.4.4. Estimation des ressources pour les activités
 - 2.4.5. Estimation de la durée des activités
 - 2.4.6. Élaboration du calendrier et calcul du chemin critique
 - 2.4.7. Contrôle des horaires
- 2.5. Élaboration du budget et réponse aux risques
 - 2.5.1. Estimation des coûts
 - 2.5.2. Élaboration du budget et de la courbe en S
 - 2.5.3. Contrôle des coûts et méthode de la valeur acquise
 - 2.5.4. Concepts de risque
 - 2.5.5. Comment faire une analyse de risque
 - 2.5.6. L'élaboration du plan d'intervention
- 2.6. Gestion de la qualité
 - 2.6.1. Planification de la qualité
 - 2.6.2. Assurance de la qualité
 - 2.6.3. Contrôle de la qualité
 - 2.6.4. Concepts statistiques fondamentaux
 - 2.6.5. Outils de la gestion de la qualité
- 2.7. Communication et ressources humaines
 - 2.7.1. Planification de la gestion des communications
 - 2.7.2. Analyse des besoins en communication
 - 2.7.3. Technologie des communications
 - 2.7.4. Modèles de communication
 - 2.7.5. Méthodes de communication

- 2.7.6. Plan de gestion de la communication
- 2.7.7. Gestion des communications
- 2.7.8. Gestion des ressources humaines
- 2.7.9. Principaux acteurs et leurs rôles dans les projets
- 2.7.10. Types d'organisations
- 2.7.11. Organisation par projet
- 2.7.12. L'équipe de travail
- 2.8. Approvisionnement
 - 2.8.1. Le processus de passation de marchés
 - 2.8.2. Planification
 - 2.8.3. Recherche de fournisseurs et appels d'offres
 - 2.8.4. Attribution du contrat
 - 2.8.5. Administration du contrat
 - 2.8.6. Contrats
 - 2.8.7. Types de contrats
 - 2.8.8. Négociation de contrats
- 2.9. Exécution, suivi et contrôle et clôture
 - 2.9.1. Les groupes de processus
 - 2.9.2. Mise en œuvre du projet
 - 2.9.3. Suivi et contrôle des projets
 - 2.9.4. Clôture du projet
- 2.10. Responsabilité professionnelle
 - 2.10.1. Responsabilité professionnelle
 - 2.10.2. Caractéristiques de la responsabilité sociale et professionnelle
 - 2.10.3. Code d'éthique du chef de projet
 - 2.10.4. Responsabilité vs. PMP®
 - 2.10.5. Exemples de responsabilité
 - 2.10.6. Avantages de la professionnalisation

Module 3. Plateformes de développement du Logiciel

- 3.1. Introduction au développement d'applications
 - 3.1.1. Applications de bureau
 - 3.1.2. Langages de programmation
 - 3.1.3. Environnements de développement intégrés
 - 3.1.4. Applications web
 - 3.1.5. Applications mobiles
 - 3.1.6. Applications du Cloud
- 3.2. Développement d'applications et d'interfaces graphiques en Java
 - 3.2.1. Environnements de développement intégrés pour Java
 - 3.2.2. Principaux IDE pour Java
 - 3.2.3. Introduction à la plate-forme de développement Eclipse
 - 3.2.4. Introduction à la plate-forme de développement NetBeans
 - 3.2.5. Modèle Vue-Contrôleur pour les interfaces utilisateur graphiques
 - 3.2.6. Conception d'une interface graphique dans Eclipse
 - 3.2.7. Conception d'une interface graphique dans NetBeans
- 3.3. Débogage et tests en Java
 - 3.3.1. Test et débogage des programmes Java
 - 3.3.2. Débogage dans Eclipse
 - 3.3.3. Débogage dans NetBeans
- 3.4. Développement d'applications et d'interfaces graphiques en .NET
 - 3.4.1. Net Framework
 - 3.4.2. Composants de la plate-forme de développement .NET
 - 3.4.3. Visual Studio .NET
 - 3.4.4. Outils .NET pour GUI
 - 3.4.5. GUI avec Windows Presentation Foundation
 - 3.4.6. Débogage et compilation d'une application WPF

- 3.5. Programmation pour les réseaux .NET
 - 3.5.1. Introduction à la programmation en réseau .NET
 - 3.5.2. Requêtes et réponses en .NET
 - 3.5.3. Utilisation de protocoles d'application dans .NET
 - 3.5.4. Sécurité dans la programmation de réseaux .NET
- 3.6. Environnements de développement d'applications mobiles
 - 3.6.1. Applications mobiles
 - 3.6.2. Applications mobiles Android
 - 3.6.3. Étapes du développement d'Android
 - 3.6.4. L'IDE Android Studio
- 3.7. Développement d'applications dans l'environnement Android Studio
 - 3.7.1. Installer et lancer Android Studio
 - 3.7.2. Exécution d'une application Android
 - 3.7.3. Développement d'une interface graphique dans Android Studio
 - 3.7.4. Démarrer des activités dans Android Studio
- 3.8. Débogage et publication d'applications Android
 - 3.8.1. Débogage d'une application dans Android Studio
 - 3.8.2. Mémoriser des applications dans Android Studio
 - 3.8.3. Publication d'une application sur Google Play
- 3.9. Développement d'applications pour le nuage
 - 3.9.1. Cloud computing
 - 3.9.2. Niveaux des nuages: SaaS, PaaS, IaaS
 - 3.9.3. Principales plates-formes de développement en nuage
 - 3.9.4. Références bibliographiques
- 3.10. Introduction à Google Cloud Platform
 - 3.10.1. Principes de base de Google Cloud Platform
 - 3.10.2. Services Google Cloud Platform
 - 3.10.3. Outils de Google Cloud Platform

Module 4. Informatique client web

- 4.1. Introduction à HTML
 - 4.1.1. Structure d'un document
 - 4.1.2. Couleur
 - 4.1.3. Texte
 - 4.1.4. Liens hypertextes
 - 4.1.5. Images
 - 4.1.6. Listes
 - 4.1.7. Tableaux
 - 4.1.8. Cadres (frames)
 - 4.1.9. Formulaires
 - 4.1.10. Éléments spécifiques aux technologies mobiles
 - 4.1.11. Articles désaffectés
- 4.2. Feuilles de style web (CSS)
 - 4.2.1. Éléments et structure d'une feuille de style
 - 4.2.1.1. Création de feuilles de style
 - 4.2.1.2. Application des styles. Sélecteurs
 - 4.2.1.3. Héritage de style et cascade
 - 4.2.1.4. Mise en page à l'aide de styles
 - 4.2.1.5. Structure de la page à l'aide de styles. Le modèle de la boîte
 - 4.2.2. Conception de styles pour différents appareils
 - 4.2.3. Types de feuilles de style: statiques et dynamiques. Pseudo-classes
 - 4.2.4. Bonnes pratiques dans l'utilisation des feuilles de style
- 4.3. Introduction et histoire de JavaScript
 - 4.3.1. Introduction
 - 4.3.2. Histoire de JavaScript
 - 4.3.3. Environnement de développement à utiliser

- 4.4. Notions de base de la programmation web
 - 4.4.1. Syntaxe de base de JavaScript
 - 4.4.2. Types de données primitives et opérateurs
 - 4.4.3. Variables et champs d'application
 - 4.4.4. Chaînes de texte et template literals
 - 4.4.5. Nombres et booléens
 - 4.4.6. Comparaisons
- 4.5. Structures complexes en JavaScript
 - 4.5.1. Vecteurs ou arrays et objets
 - 4.5.2. Sets
 - 4.5.3. Cartes
 - 4.5.4. Disjonctions
 - 4.5.5. Boucles
- 4.6. Fonctions et objectifs
 - 4.6.1. Définition et invocation des fonctions
 - 4.6.2. Argumentation
 - 4.6.3. Fonctions de la flèche
 - 4.6.4. Fonctions de rappel ou callback
 - 4.6.5. Fonctions d'ordre supérieur
 - 4.6.6. Objets littéraux
 - 4.6.7. L'objet this
 - 4.6.8. Les objets en tant qu'espaces de noms: l'objet Math et l'objet Date
- 4.7. Le modèle objet du document (DOM)
 - 4.7.1. Qu'est-ce que le DOM?
 - 4.7.2. Un peu d'histoire
 - 4.7.3. Navigation et obtention d'éléments
 - 4.7.4. Un DOM virtuel avec JSDOM
 - 4.7.5. Sélecteurs de requête ou query selectors
 - 4.7.6. Navigation par les propriétés
 - 4.7.7. Attribution d'attributs aux éléments
 - 4.7.8. Création et modification de nœuds
 - 4.7.9. Mise à jour du style des éléments du DOM
- 4.8. Développement web moderne
 - 4.8.1. Flux événementiel et listeners
 - 4.8.2. Toolkits web modernes et systèmes d'alignement
 - 4.8.3. Mode strict JavaScript
 - 4.8.4. Plus sur les fonctions
 - 4.8.5. Promesses et fonctions asynchrones
 - 4.8.6. Fermetures
 - 4.8.7. Programmation fonctionnelle
 - 4.8.8. POO en JavaScript
- 4.9. Utilisabilité du web
 - 4.9.1. Introduction à l'utilisabilité
 - 4.9.2. Définition de l'utilisabilité
 - 4.9.3. Importance de la conception de sites web centrés sur l'utilisateur
 - 4.9.4. Différences entre l'accessibilité et la facilité d'utilisation
 - 4.9.5. Avantages et problèmes liés à la combinaison de l'accessibilité et de la facilité d'utilisation
 - 4.9.6. Avantages et difficultés de la mise en œuvre de sites web utilisables
 - 4.9.7. Méthodes d'utilisabilité
 - 4.9.8. Analyse des besoins des utilisateurs
 - 4.9.9. Principes du design conceptuel. Création de prototypes orientés vers l'utilisateur
 - 4.9.10. Lignes directrices pour la création de sites web utilisables
 - 4.9.10.1. Lignes directrices d'utilisabilité de Jakob Nielsen
 - 4.9.10.2. Lignes directrices d'utilisabilité par Bruce Tognazzini
 - 4.9.11. Évaluation de la convivialité
- 4.10. Accessibilité du web
 - 4.10.1. Introduction
 - 4.10.2. Définition de l'accessibilité du web
 - 4.10.3. Types de handicaps
 - 4.10.3.1. Handicaps temporaires ou permanents
 - 4.10.3.2. Déficience visuelle
 - 4.10.3.3. Déficience auditive
 - 4.10.3.4. Handicaps moteurs

- 4.10.3.5. Handicaps neurologiques ou cognitifs
- 4.10.3.6. Difficultés liées au vieillissement
- 4.10.3.7. Contraintes liées à l'environnement
- 4.10.3.8. Obstacles à l'accès au web
- 4.10.4. Aides techniques et produits d'assistance pour surmonter les obstacles
 - 4.10.4.1. Aides pour les aveugles
 - 4.10.4.2. Aides pour les personnes malvoyantes
 - 4.10.4.3. Aides pour les daltoniens
 - 4.10.4.4. Aides pour les personnes malentendantes
 - 4.10.4.5. Aides pour les personnes handicapées moteur
 - 4.10.4.6. Aides pour les personnes souffrant de handicaps cognitifs et neurologiques
- 4.10.5. Avantages et difficultés de la mise en œuvre de l'accessibilité du web
- 4.10.6. Réglementation et normes en matière d'accessibilité du web
- 4.10.7. Organismes de réglementation de l'accessibilité du web
- 4.10.8. Comparaison des normes et des standards
- 4.10.9. Lignes directrices pour le respect des règlements et des normes
 - 4.10.9.1. Description des principales lignes directrices (images, liens, vidéos, etc.)
 - 4.10.9.2. Lignes directrices pour une navigation accessible
 - 4.10.9.2.1. Perceptibilité
 - 4.10.9.2.2. Opérationnalité
 - 4.10.9.2.3. Compréhensibilité
 - 4.10.9.2.4. Robustesse
- 4.10.10. Description du processus de mise en conformité de l'accessibilité du web
- 4.10.11. Niveaux de conformité
- 4.10.12. Critères de conformité
- 4.10.13. Exigences de conformité
- 4.10.14. Méthodologie d'évaluation de l'accessibilité des sites web

Module 5. Informatique des serveurs web

- 5.1. Introduction à la programmation côté serveur: PHP
 - 5.1.1. Principes de base de la programmation côté serveur
 - 5.1.2. Syntaxe de base de PHP
 - 5.1.3. Génération de contenu HTML avec PHP
 - 5.1.4. Environnements de développement et de test: XAMPP
- 5.2. PHP avancé
 - 5.2.1. Structures de contrôle avec PHP
 - 5.2.2. Fonctions de PHP
 - 5.2.3. Gestion des arrays en PHP
 - 5.2.4. Traitement des chaînes de caractères avec PHP
 - 5.2.5. Orientation vers l'objet en PHP
- 5.3. Modèles de données
 - 5.3.1. Concept de données. Cycle de vie des données
 - 5.3.2. Types de données
 - 5.3.2.1. Bases
 - 5.3.2.2. Enregistrements
 - 5.3.2.3. Dynamisme
- 5.4. Le modèle relationnel
 - 5.4.1. Description
 - 5.4.2. Entités et types d'entités
 - 5.4.3. Éléments de données. Attributs
 - 5.4.4. Relations: types, sous-types, cardinalité
 - 5.4.5. Clés. Types de clés
 - 5.4.6. Normalisation Formes normales
- 5.5. Construction du modèle logique de données
 - 5.5.1. Spécification des tableaux
 - 5.5.2. Définition des colonnes
 - 5.5.3. Spécifications clés
 - 5.5.4. Conversion en formes normales. Dépendances

- 5.6. Le modèle physique de données. Fichiers de données
 - 5.6.1. Description des fichiers de données
 - 5.6.2. Types de fichiers
 - 5.6.3. Modes d'accès
 - 5.6.4. Organisation des dossiers
- 5.7. Accès aux bases de données à partir de PHP
 - 5.7.1. Introduction à MariaDB
 - 5.7.2. Travailler avec une base de données MariaDB: le langage SQL
 - 5.7.3. Accéder à la base de données MariaDB depuis PHP
 - 5.7.4. Introduction à MySQL
 - 5.7.5. Travailler avec une base de données MySQL: le langage SQL
 - 5.7.6. Accéder à la base de données MySQL depuis PHP
- 5.8. Interaction avec le client à partir de PHP
 - 5.8.1. Formulaires PHP
 - 5.8.2. Cookies
 - 5.8.3. Gestion des sessions
- 5.9. Architecture des applications web
 - 5.9.1. Le modèle Modèle Vue Contrôleur
 - 5.9.2. Contrôleur
 - 5.9.3. Modèle
 - 5.9.4. Afficher
- 5.10. Introduction aux services web
 - 5.10.1. Introduction à XML
 - 5.10.2. Architectures orientées services (SOA): services web
 - 5.10.3. Création de services web SOAP et REST
 - 5.10.4. Le protocole SOAP
 - 5.10.5. Le protocole REST

Module 6. Gestion de la sécurité

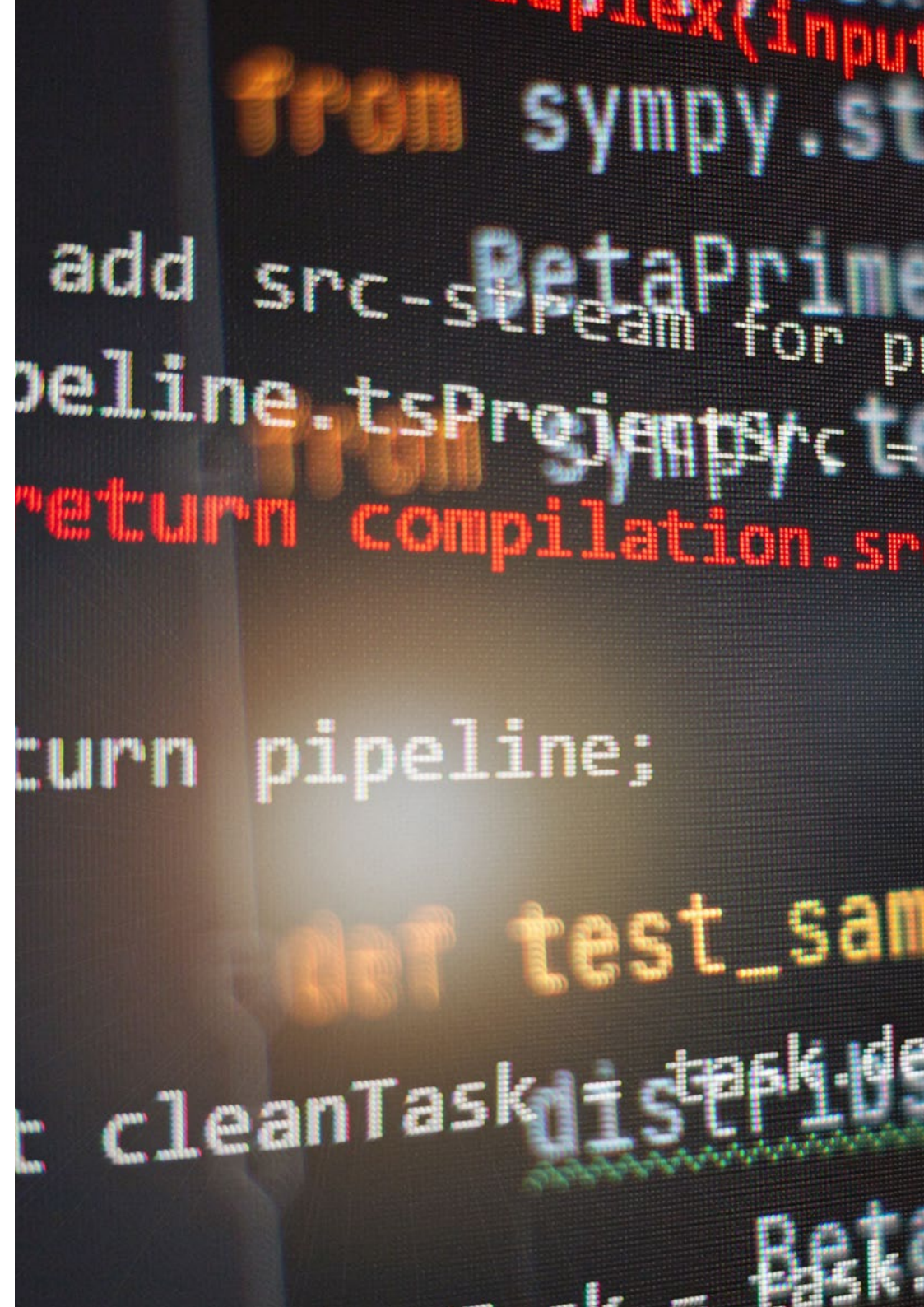
- 6.1. La sécurité de l'information
 - 6.1.1. Introduction
 - 6.1.2. La sécurité de l'information implique la confidentialité, l'intégrité et la disponibilité
 - 6.1.3. La sécurité est une question économique
 - 6.1.4. La sécurité est un processus
 - 6.1.5. La classification des informations
 - 6.1.6. La sécurité de l'information implique une gestion des risques
 - 6.1.7. La sécurité s'articule avec les contrôles de sécurité
 - 6.1.8. La sécurité est à la fois physique et logique
 - 6.1.9. La sécurité implique des personnes
- 6.2. Le professionnel de la sécurité de l'information
 - 6.2.1. Introduction
 - 6.2.2. La sécurité de l'information en tant que profession
 - 6.2.3. Certifications (ISC)2
 - 6.2.4. La norme ISO 27001
 - 6.2.5. Bonnes pratiques de sécurité dans la gestion des services IT
 - 6.2.6. Modèles de maturité pour la sécurité de l'information
 - 6.2.7. Autres certifications, normes et ressources professionnelles
- 6.3. Contrôle d'accès
 - 6.3.1. Introduction
 - 6.3.2. Exigences en matière de contrôle d'accès
 - 6.3.3. Mécanismes d'authentification
 - 6.3.4. Méthodes d'autorisation
 - 6.3.5. Comptabilité d'accès et audit
 - 6.3.6. Technologies « Triple A »

- 6.4. Programmes, processus et politiques en matière de sécurité de l'information
 - 6.4.1. Introduction
 - 6.4.2. Programmes de gestion de la sécurité
 - 6.4.3. Gestion des risques
 - 6.4.4. Conception de la politique de sécurité
- 6.5. Plans de continuité des activités
 - 6.5.1. Introduction aux PCN
 - 6.5.2. Phases I et II
 - 6.5.3. Phases III et IV
 - 6.5.4. Maintenance du PCN
- 6.6. Procédures pour la protection adéquate de l'entreprise
 - 6.6.1. Réseaux DMZ
 - 6.6.2. Systèmes de détection d'intrusion
 - 6.6.3. Listes de contrôle d'accès
 - 6.6.4. Apprendre de l'attaquant: Honeypot
- 6.7. Architecture de sécurité. Prévention
 - 6.7.1. Aperçu général Activités et modèle de couche
 - 6.7.2. Défense du périmètre (pare-feu, WAF, IPS, etc.)
 - 6.7.3. Défense des points finaux (équipements, serveurs et services)
- 6.8. Architecture de sécurité. Détection
 - 6.8.1. Aperçu de la détection et de la surveillance
 - 6.8.2. Logs, rupture de trafic cryptée, enregistrement et Siems
 - 6.8.3. Alertes et renseignements
- 6.9. Architecture de sécurité. Réaction
 - 6.9.1. Réaction. Produits, services et ressources
 - 6.9.2. Gestion des Incidents de
 - 6.9.3. CERTS et CSIRT
- 6.10. Architecture de sécurité. Récupération
 - 6.10.1. Résilience, Concepts, Exigences Commerciales et Réglementations
 - 6.10.2. Solutions IT de Résilience
 - 6.10.3. Gestion et Gouvernance des Crises

Module 7. Sécurité des systèmes d'information

- 7.1. Un aperçu de la sécurité, de la cryptographie et des cryptanalyses classiques
 - 7.1.1. Sécurité informatique: Perspective historique
 - 7.1.2. Mais qu'entend-on exactement par sécurité?
 - 7.1.3. Histoire de la cryptographie
 - 7.1.4. Chiffres de substitution
 - 7.1.5. Étude de cas: La machine Enigma
- 7.2. Cryptographie symétrique
 - 7.2.1. Introduction et terminologie de base
 - 7.2.2. Cryptage symétrique
 - 7.2.3. Modes d'opération
 - 7.2.4. DES
 - 7.2.5. La nouvelle norme AES
 - 7.2.6. Cryptage de flux
 - 7.2.7. Cryptanalyse
- 7.3. Cryptographie asymétrique
 - 7.3.1. Origines de la cryptographie à clé publique
 - 7.3.2. Concepts de base et fonctionnement
 - 7.3.3. L'algorithme RSA
 - 7.3.4. Certificats numériques
 - 7.3.5. Stockage et gestion des clés
- 7.4. Attaques de réseau
 - 7.4.1. Menaces et attaques contre les réseaux
 - 7.4.2. Énumération
 - 7.4.3. Interception du trafic: sniffers
 - 7.4.4. Attaques par déni de service
 - 7.4.5. Attaques par empoisonnement ARP

- 7.5. Architectures de sécurité
 - 7.5.1. Architectures de sécurité traditionnel
 - 7.5.2. Secure Socket Layer: SSL
 - 7.5.3. Protocole SSH
 - 7.5.4. Réseaux Privés Virtuels (VPN)
 - 7.5.5. Mécanismes de protection des unités de stockage externes
 - 7.5.6. Mécanismes de protection du matériel
- 7.6. Techniques de protection des systèmes et développement de code sécurisé
 - 7.6.1. Sécurité Opérationnelle
 - 7.6.2. Ressources et contrôles
 - 7.6.3. Suivi
 - 7.6.4. Systèmes de détection des intrusions
 - 7.6.5. IDS de l'hôte
 - 7.6.6. IDS réseau
 - 7.6.7. IDS basé sur les signatures
 - 7.6.8. Systèmes de leurres
 - 7.6.9. Principes de base de la sécurité dans le développement du code
 - 7.6.10. Gestion des pannes
 - 7.6.11. Ennemi public numéro 1: les dépassements de tampon (Buffer Overflows)
 - 7.6.12. Botches cryptographiques
- 7.7. Botnets et spam
 - 7.7.1. Origine du problème
 - 7.7.2. Processus de spam
 - 7.7.3. Envoi de spam
 - 7.7.4. Affinement des listes de diffusion
 - 7.7.5. Techniques de protection
 - 7.7.6. Service anti- offert par des tiers
 - 7.7.7. Étude de cas
 - 7.7.8. Spam exotique



- 7.8. Audit et attaques du Web
 - 7.8.1. Collecte d'informations
 - 7.8.2. Techniques d'attaque
 - 7.8.3. Outils
- 7.9. Malware et code malveillant
 - 7.9.1. Qu'est-ce qu'un malware?
 - 7.9.2. Types de malware
 - 7.9.3. Virus
 - 7.9.4. Cryptovirus
 - 7.9.5. Vers
 - 7.9.6. Adware
 - 7.9.7. Spyware
 - 7.9.8. Canulars
 - 7.9.9. Pishing
 - 7.9.10. Trojans
 - 7.9.11. L'économie de Malware
 - 7.9.12. Solutions possibles
- 7.10. Analyse médico-légale
 - 7.10.1. Collecte des preuves
 - 7.10.2. Analyse des preuves
 - 7.10.3. Techniques anti-forensic
 - 7.10.4. Étude de cas pratique

Module 8. Sécurité du Software

- 8.1. Questions relatives à la sécurité des Softwares
 - 8.1.1. Introduction au problème de la sécurité des Softwares
 - 8.1.2. Les vulnérabilités et leur classification
 - 8.1.3. Propriétés des Softwares sécurisés
 - 8.1.4. Références
- 8.2. Principes de conception de la sécurité des Softwares
 - 8.2.1. Introduction
 - 8.2.2. Principes de conception de la sécurité des Softwares
 - 8.2.3. Types de S-SDLC
 - 8.2.4. Sécurité des Softwares dans les phases S-SDLC
 - 8.2.5. Méthodologies et normes
 - 8.2.6. Références
- 8.3. Sécurité dans le cycle de vie du Software dans les phases d'exigences et de conception
 - 8.3.1. Introduction
 - 8.3.2. Modélisation des attaques
 - 8.3.3. Cas d'abus
 - 8.3.4. Ingénierie des exigences de sécurité
 - 8.3.5. Analyse des risques. Architectural
 - 8.3.6. Modèles de conception
 - 8.3.7. Références
- 8.4. Sécurité du cycle de vie des Softwares dans les phases de codage, de test et d'exploitation
 - 8.4.1. Introduction
 - 8.4.2. Tests de sécurité fondés sur le risque
 - 8.4.3. Examen du code
 - 8.4.4. Test de pénétration
 - 8.4.5. Opérations de sécurité
 - 8.4.6. Examen externe
 - 8.4.7. Références
- 8.5. Applications de codage sécurisé I
 - 8.5.1. Introduction
 - 8.5.2. Pratiques de codage sécurisées
 - 8.5.3. Traitement et validation des entrées
 - 8.5.4. Débordement de mémoire
 - 8.5.5. Références
- 8.6. Applications de codage sécurisé II
 - 8.6.1. Introduction
 - 8.6.2. Integers overflows, erreurs de troncature et problèmes de conversion de type entre entiers
 - 8.6.3. Erreurs et exceptions
 - 8.6.4. Vie privée et confidentialité
 - 8.6.5. Programmes privilégiés
 - 8.6.6. Références
- 8.7. La sécurité dans le développement et dans le cloud
 - 8.7.1. La sécurité dans le développement ; méthodologie et pratique
 - 8.7.2. Modèles PaaS, IaaS, CaaS et SaaS
 - 8.7.3. Sécurité dans le nuage et pour les services Cloud
- 8.8. Orchestration et automatisation de la sécurité (SOAR)
 - 8.8.1. Complexité du traitement manuel ; nécessité d'automatiser les tâches
 - 8.8.2. Produits et services
 - 8.8.3. Architecture du SOAR
- 8.9. La sécurité dans le Télétravail
 - 8.9.1. Besoin et scénarios
 - 8.9.2. Produits et services
 - 8.9.3. La sécurité dans le télétravail

Module 9. Qualité et audit des systèmes d'information

- 9.1. Introduction aux Systèmes de Gestion de la Sécurité de l'Information
 - 9.1.1. Principes fondamentaux du SMSI
 - 9.1.2. Règles d'or du SMSI
 - 9.1.3. Rôle de l'audit informatique dans les SGSI
- 9.2. La planification dans la gestion de la sécurité
 - 9.2.1. Concepts liés à la gestion de la sécurité
 - 9.2.2. Classification de l'information: objectifs, concepts et rôles
 - 9.2.3. Mise en œuvre des politiques de sécurité: Politiques de sécurité, normes et procédures
 - 9.2.4. Gestion des risques: Principes et analyse des risques liés aux actifs informationnels
- 9.3. Principaux mécanismes de protection des actifs informationnels (I)
 - 9.3.1. Aperçu des principaux outils cryptographiques pour la protection de la triade CID
 - 9.3.2. Prise en compte de la vie privée, de l'anonymat et des exigences de gestion de la traçabilité des utilisateurs
- 9.4. Principaux mécanismes de protection des actifs informationnels (II)
 - 9.4.1. Sécurité des communications: protocoles, dispositifs et architectures de sécurité
 - 9.4.2. Sécurité des systèmes d'exploitation
- 9.5. Contrôle interne des SGSI
 - 9.5.1. Taxonomie des contrôles du SMSI: contrôles administratifs, logiques et physiques
 - 9.5.2. Classification des contrôles en fonction de la manière dont la menace est traitée: Contrôles pour la prévention, la détection et la correction des menaces
 - 9.5.3. Mise en œuvre de systèmes de contrôle interne dans les SGSI
- 9.6. Types d'audits
 - 9.6.1. Différence entre l'audit et le contrôle interne
 - 9.6.2. Audit interne et externe
 - 9.6.3. Classification de l'audit en fonction de l'objectif et du type d'analyse
- 9.7. Scénariste et scénario: Sujet et objet protégés par le droit de la Propriété Intellectuelle
 - 9.7.1. Introduction aux tests de pénétration et à l'analyse médico-légale
 - 9.7.2. Définition et pertinence des concepts de fingerprinting et footprinting
- 9.8. Analyse de vulnérabilité et surveillance du trafic réseau
 - 9.8.1. Outils pour l'analyse de la vulnérabilité des systèmes
 - 9.8.2. Principales vulnérabilités dans le contexte des applications web
 - 9.8.3. Analyse des protocoles de communication

- 9.9. Le processus d'audit informatique
 - 9.9.1. Concept de cycle de vie du développement des systèmes
 - 9.9.2. Surveillance des activités et des processus: collecte et traitement des données
 - 9.9.3. Méthodologies d'audit informatique
 - 9.9.4. Le processus d'audit informatique
 - 9.9.5. Identification des principales infractions et délits dans le contexte informatique
 - 9.9.6. Enquête sur la criminalité informatique: introduction à l'analyse forensique et à sa relation avec l'audit informatique
- 9.10. Plans de continuité des activités et de reprise après sinistre
 - 9.10.1. Définition du plan de continuité d'activité et du concept d'interruption d'activité
 - 9.10.2. Recommandation du NIST sur les plans de continuité des activités
 - 9.10.3. Plan de reprise après sinistre
 - 9.10.4. Processus de plan de reprise après sinistre

Module 10. Administration des serveurs web

- 10.1. Introduction aux serveurs web
 - 10.1.1. Qu'est-ce qu'un serveur web?
 - 10.1.2. Architecture et fonctionnement d'un serveur web
 - 10.1.3. Ressources et contenu d'un serveur web
 - 10.1.4. Serveurs d'application
 - 10.1.5. Serveurs proxy
 - 10.1.6. Principaux serveurs web du marché
 - 10.1.7. Statistiques d'utilisation du serveur web
 - 10.1.8. Sécurité des serveurs web
 - 10.1.9. Équilibrage de la charge sur les serveurs web
 - 10.1.10. Références
- 10.2. Traitement du protocole HTTP
 - 10.2.1. Fonctionnement et structure
 - 10.2.2. Description des demandes ou des request methods
 - 10.2.3. Codes d'état
 - 10.2.4. En-têtes
 - 10.2.5. Codage du contenu. Pages de code
 - 10.2.6. Effectuer des requêtes HTTP sur l'internet à l'aide d'un proxy, de livehttpheaders ou d'une méthode similaire, en analysant le protocole utilisé

- 10.3. Description des architectures distribuées multi-serveurs
 - 10.3.1. Modèle à trois couches
 - 10.3.2. Tolérance aux pannes
 - 10.3.3. Partage des charges
 - 10.3.4. Stockage de l'état de la session
 - 10.3.5. Magasins de stockage
- 10.4. Internet Information Services (IIS)
 - 10.4.1. Qu'est-ce que l'IIS?
 - 10.4.2. Histoire et évolution de l'IIS
 - 10.4.3. Principaux avantages et caractéristiques d'IIS7 et des versions ultérieures
 - 10.4.4. Architecture IIS7 et ultérieures
- 10.5. Installation, administration et configuration de IIS
 - 10.5.1. Préambule
 - 10.5.2. Installation d'Internet Information Services (IIS)
 - 10.5.3. Outils d'administration IIS
 - 10.5.4. Création, configuration et administration de sites web
 - 10.5.5. Installation et gestion des extensions dans IIS
- 10.6. Sécurité avancée dans IIS
 - 10.6.1. Préambule
 - 10.6.2. Authentification, autorisation et contrôle d'accès dans IIS
 - 10.6.3. Mise en place d'un site web sécurisé sur IIS avec SSL
 - 10.6.4. Politiques de sécurité mises en œuvre dans IIS 8.x
- 10.7. Introduction à Apache
 - 10.7.1. Qu'est-ce qu'Apache?
 - 10.7.2. Principaux avantages d'Apache
 - 10.7.3. Principales caractéristiques d'Apache
 - 10.7.4. Architecture
- 10.8. Installation et configuration d'Apache
 - 10.8.1. Installation initiale d'Apache
 - 10.8.2. Configuration d'Apache

- 10.9. Installation et configuration des différents modules d'Apache
 - 10.9.1. Installation des modules Apache
 - 10.9.2. Types de modules
 - 10.9.3. Configuration sécurisée d'Apache
- 10.10. Sécurité avancée
 - 10.10.1. Authentification, autorisation et contrôle d'accès
 - 10.10.2. Méthodes d'authentification
 - 10.10.3. Configuration sécurisée d'Apache avec SSL

Module 11. Sécurité des applications online

- 11.1. Vulnérabilités et problèmes de sécurité dans les applications online
 - 11.1.1. Introduction à la sécurité dans les applications online
 - 11.1.2. Les failles de sécurité dans la conception des applications web
 - 11.1.3. Les failles de sécurité dans la mise en œuvre des applications web
 - 11.1.4. Les failles de sécurité dans le déploiement des applications web
 - 11.1.5. Listes officielles de failles de sécurité
- 11.2. Politiques et normes relatives à la sécurité des applications online
 - 11.2.1. Piliers de la sécurité des applications online
 - 11.2.2. Politique de sécurité
 - 11.2.3. Système de gestion de la sécurité de l'information
 - 11.2.4. Cycle de développement sécurisé des Logiciels
 - 11.2.5. Normes pour la sécurité des applications
- 11.3. La sécurité dans la conception des applications web
 - 11.3.1. Introduction à la sécurité des applications web
 - 11.3.2. La sécurité dans la conception des applications web
- 11.4. Tester la sécurité online et la sûreté des applications web
 - 11.4.1. Analyse et test de la sécurité des applications web
 - 11.4.2. Sécurité dans le déploiement et la production d'applications web
- 11.5. Sécurité des services web
 - 11.5.1. Introduction à la sécurité des applications web
 - 11.5.2. Fonctions et technologies de sécurité des services web

- 11.6. Tester la sécurité et la protection en ligne des services web
 - 11.6.1. Évaluation de la sécurité des services web
 - 11.6.2. Protection en ligne. Firewalls et gateways XML
- 11.7. Hacking éthique, malware y forensic
 - 11.7.1. Piratage éthique
 - 11.7.2. Analyse de Malware
 - 11.7.3. Analyse Forense
- 11.8. Résolution d'incidents pour les services web
 - 11.8.1. Suivi
 - 11.8.2. Outils de mesure de la performance
 - 11.8.3. Mesures de confinement
 - 11.8.4. Analyse des causes profondes
 - 11.8.5. Gestion proactive des problèmes
- 11.9. Bonnes pratiques pour assurer la sécurité des applications
 - 11.9.1. Manuel de bonnes pratiques pour le développement d'applications online
 - 11.9.2. Manuel de bonnes pratiques pour la mise en œuvre des applications online
- 11.10. Erreurs courantes qui compromettent la sécurité des applications
 - 11.10.1. Erreurs de développement courantes
 - 11.10.2. Erreurs courantes en matière d'hébergement
 - 11.10.3. Erreurs courantes dans la production

Module 12. Ingénierie Software

- 12.1. Introduction au Génie Logiciel et à la modélisation
 - 12.1.1. La nature des Logiciels
 - 12.1.2. La nature unique des webApps
 - 12.1.3. Ingénierie Software
 - 12.1.4. Le processus Logiciel
 - 12.1.5. La pratique du Génie Logiciel
 - 12.1.6. Mythes sur les Logiciels
 - 12.1.7. Comment tout commence
 - 12.1.8. Concepts orientés objet
 - 12.1.9. Concepts orientés objet
- 12.2. Le processus Logiciel
 - 12.2.1. Un modèle général de processus
 - 12.2.2. Modèles de processus prescriptifs
 - 12.2.3. Modèles de processus spécialisés
 - 12.2.4. Le processus unifié
 - 12.2.5. Modèles de processus personnels et d'équipe
 - 12.2.6. Qu'est-ce que l'agilité?
 - 12.2.7. Qu'est-ce qu'un processus agile?
 - 12.2.8. Scrum
 - 12.2.9. Boîte à outils du processus Agile
- 12.3. Principes guidant la pratique du Génie Logiciel
 - 12.3.1. Principes guidant le processus
 - 12.3.2. Principes guidant la pratique
 - 12.3.3. Principes de la communication
 - 12.3.4. Principes de planification
 - 12.3.5. Principes de modélisation
 - 12.3.6. Principes de construction
 - 12.3.7. Principes de déploiement
- 12.4. Comprendre les besoins
 - 12.4.1. Ingénierie des exigences
 - 12.4.2. Établir la base
 - 12.4.3. Détermination des besoins
 - 12.4.4. Développer des cas d'utilisation
 - 12.4.5. Élaboration du modèle d'exigences
 - 12.4.6. Négociation des exigences
 - 12.4.7. Validation des exigences
- 12.5. Modélisation des exigences: Scénarios, informations et types d'analyse
 - 12.5.1. Analyse des besoins
 - 12.5.2. Modélisation basée sur des scénarios
 - 12.5.3. Modèles UML fournissant le cas d'utilisation
 - 12.5.4. Concepts de modélisation des données
 - 12.5.5. Modélisation basée sur les classes
 - 12.5.6. Diagrammes de classes

- 12.6. Modélisation des exigences: Flux, comportement et modèles
 - 12.6.1. Stratégies de modélisation des exigences
 - 12.6.2. Modélisation orientée flux
 - 12.6.3. Diagrammes d'état
 - 12.6.4. Création d'un modèle comportemental
 - 12.6.5. Diagrammes de séquence
 - 12.6.6. Diagrammes de communication
 - 12.6.7. Modèles pour la modélisation des exigences
- 12.7. Concepts de design
 - 12.7.1. Conception dans le contexte de l'Ingénierie Logicielle
 - 12.7.2. Le processus de conception
 - 12.7.3. Concepts de design
 - 12.7.4. Concepts de conception orientée objet
 - 12.7.5. Le modèle de conception
- 12.8. Architecture de conception
 - 12.8.1. Architecture des Logiciels
 - 12.8.2. Genres architecturaux
 - 12.8.3. Styles architecturaux
 - 12.8.4. Conception architecturale
 - 12.8.5. Évolution des conceptions alternatives de l'architecture
 - 12.8.6. Cartographie de l'architecture à l'aide du flux de données
- 12.9. Conception au niveau des composants et des modèles
 - 12.9.1. Qu'est-ce qu'un composant?
 - 12.9.2. Conception de composants basée sur les classes
 - 12.9.3. Réalisation de la conception au niveau des composants
 - 12.9.4. Conception traditionnelle des composants
 - 12.9.5. Développement basé sur les composants
 - 12.9.6. Modèles de conception
 - 12.9.7. Conception de Logiciels basée sur des modèles
 - 12.9.8. Modèles architecturaux
 - 12.9.9. Patrons de conception au niveau des composants
 - 12.9.10. Modèles de conception d'interface utilisateur

- 12.10. Qualité des logiciels et gestion de projet
 - 12.10.1. Qualité
 - 12.10.1.1. Qualité du Software
 - 12.10.2. Le dilemme de la qualité des Logiciels
 - 12.10.3. Atteindre la qualité des Logiciels
 - 12.10.4. Assurance qualité des logiciels
 - 12.10.5. Le spectre de la gestion
 - 12.10.6. Personnel
 - 12.10.7. Le produit
 - 12.10.8. Le processus
 - 12.10.9. Le projet
 - 12.10.10. Principes et pratiques

Module 13. Génie Logiciel avancée

- 13.1. Programmation extrême
 - 13.1.1. Justification et présentation de XP
 - 13.1.2. Le cycle de vie au XP
 - 13.1.3. Les cinq valeurs de base
 - 13.1.4. Les douze pratiques de base de XP
 - 13.1.5. Rôles des participants
 - 13.1.6. XP Industriel
 - 13.1.7. Évaluation critique de XP
- 13.2. Développement de Logiciels basés sur la réutilisation
 - 13.2.1. Réutilisation de Software
 - 13.2.2. Niveaux de réutilisation du code
 - 13.2.3. Techniques spécifiques de réutilisation
 - 13.2.4. Développement basé sur les composants
 - 13.2.5. Avantages et problèmes de réutilisation
 - 13.2.6. Planification de la réutilisation

- 13.3. Modèles d'architecture système et de conception Logicielle
 - 13.3.1. Conception architecturale
 - 13.3.2. Modèles architecturaux généraux
 - 13.3.3. Architectures tolérantes aux pannes
 - 13.3.4. Architectures de systèmes distribués
 - 13.3.5. Modèles de conception
 - 13.3.6. Modèles Gamma
 - 13.3.7. Modèles de conception d'interaction
- 13.4. Architecture d'application cloud
 - 13.4.1. Principes fondamentaux de Cloud Computing
 - 13.4.2. Qualité d'application cloud
 - 13.4.3. Styles d'architecture
 - 13.4.4. Modèles de conception
- 13.5. Tests de Logiciels: TDD, ATDD et BDD
 - 13.5.1. Vérification et validation du Logiciel
 - 13.5.2. Tests de Logiciels
 - 13.5.3. Test Driven Development (TDD)
 - 13.5.4. Acceptance Test Driven Development (ATDD)
 - 13.5.5. Behavior Driven Development (BDD)
 - 13.5.6. BDD et Cucumber
- 13.6. L'amélioration du processus Logiciel
 - 13.6.1. L'amélioration du processus Logiciel
 - 13.6.2. Le processus d'amélioration de processus
 - 13.6.3. Modèles de maturité
 - 13.6.4. Le modèle CMMI
 - 13.6.5. CMMI V2.0. 13.6.6. CMMI et Agile
- 13.7. La qualité du produit Logiciel: SQuaRE
 - 13.7.1. La qualité des Logiciels
 - 13.7.2. Modèles de qualité des produit Logiciels
 - 13.7.3. Famille ISO/IEC 25000
 - 13.7.4. ISO/IEC 25010: modèle et caractéristiques de qualité
 - 13.7.5. ISO/IEC 25012: qualité des données
 - 13.7.6. ISO/IEC 25020: mesure de la qualité du Logiciel
 - 13.7.7. ISO/IEC 25022, 25023 et 25024: mesures de qualité des Logiciels et des données
 - 13.7.8. ISO/IEC 25040: évaluation du Logiciel
 - 13.7.9. Processus de certification
- 13.8. Présentation de DevOps
 - 13.8.1. Concept DevOps
 - 13.8.2. Principales pratiques

Module 14. Ingénierie des exigences

- 14.1. Introduction à l'Ingénierie des exigences
 - 14.1.1. L'importance des exigences
 - 14.1.2. Concept d'exigence
 - 14.1.3. Dimensions des exigences
 - 14.1.4. Niveaux et types d'exigences
 - 14.1.5. Caractéristiques des exigences
 - 14.1.6. Ingénierie des exigences
 - 14.1.7. Le processus d'Ingénierie des exigences
 - 14.1.8. Frameworks pour l'Ingénierie des exigences
 - 14.1.9. Meilleures pratiques en matière d'ingénierie des exigences
 - 14.1.10. L'analyste d'affaires
- 14.2. Les sources des besoins
 - 14.2.1. Le réseau des exigences
 - 14.2.2. Les stakeholders
 - 14.2.3. Les exigences commerciales
 - 14.2.4. Document sur la vision et le champ d'application
- 14.3. Techniques d'élicitation des besoins
 - 14.3.1. Détermination des besoins
 - 14.3.2. Problèmes liés à la détermination des besoins
 - 14.3.3. Contextes de découverte
 - 14.3.4. Interviews
 - 14.3.5. Observation et "apprentissage"

- 14.3.6. Ethnographie
- 14.3.7. Workshops
- 14.3.8. Focus groups
- 14.3.9. Questionnaires
- 14.3.10. Brainstorming et techniques de création
- 14.3.11. Médias du groupe
- 14.3.12. Analyse des interfaces du système
- 14.3.13. Analyse de documents et "archéologie"
- 14.3.14. Cas d'utilisation et scénario
- 14.3.15. Prototypes
- 14.3.16. Ingénierie inverse
- 14.3.17. Réutilisation des exigences
- 14.3.18. Bonnes pratiques en matière d'appels d'offres
- 14.4. Besoins des utilisateurs
 - 14.4.1. Personnes
 - 14.4.2. Cas d'utilisation et histoires d'utilisateurs
 - 14.4.3. Scénario
 - 14.4.5. Types de scénarios
 - 14.4.6. Comment découvrir des scénarios
- 14.5. Techniques de Prototypage
 - 14.5.1. Prototypage
 - 14.5.2. Prototypes en fonction de leur portée
 - 14.5.3. Prototypes en fonction de leur temporalité
 - 14.5.4. La fidélité d'un prototype
 - 14.5.5. Prototypes d'interface utilisateur
 - 14.5.6. Évaluation des prototypes
- 14.6. Analyse des besoins
 - 14.6.1. Analyse des besoins
 - 14.6.2. Bonnes pratiques en matière d'analyse des exigences
 - 14.6.3. Dictionnaire de données
 - 14.6.4. Hiérarchisation des besoins
- 14.7. Documentation des exigences
 - 14.7.1. Le document de spécification des exigences
 - 14.7.2. Structure et contenu d'un BSS
 - 14.7.3. Documentation en langage naturel
 - 14.7.4. EARS: Easy Approach to Requirements Syntax
 - 14.7.5. Exigences non fonctionnelles
 - 14.7.6. Attributs et modèles de table
 - 14.7.7. Bonne pratique de spécification
- 14.8. Validation et négociation des exigences
 - 14.8.1. Validation des exigences
 - 14.8.2. Techniques de validation des exigences
 - 14.8.3. Négociation des exigences
- 14.9. Modélisation et gestion des exigences
 - 14.9.1. Modélisation des besoins
 - 14.9.2. Le point de vue de l'utilisateur
 - 14.9.3. Perspective des données
 - 14.9.4. La perspective fonctionnelle ou axée sur le flux
 - 14.9.5. La perspective comportementale
 - 14.9.6. Volatilité des exigences
 - 14.9.7. Processus de gestion des exigences
 - 14.9.8. Outils de gestion des exigences
 - 14.9.9. Bonnes pratiques en matière de gestion des exigences
- 14.10. Systèmes critiques et spécification formelle
 - 14.10.1. Systèmes critiques
 - 14.10.2. Spécification axée sur le risque
 - 14.10.3. Spécification formelle

Module 15. Processus d'Ingénierie Software

- 15.1. Cadre de l'Ingénierie Logicielle
 - 15.1.1. Caractéristiques des Logiciels
 - 15.1.2. Les principaux processus de l'Ingénierie Software
 - 15.1.3. Modèles de processus de développement de Logiciels
 - 15.1.4. Cadre de référence standard pour le processus de développement de Logiciels: Normes ISO/IEC 12207
- 15.2. Processus unifié de développement de Logiciels
 - 15.2.1. Processus Unifié
 - 15.2.2. Les dimensions du Processus Unifié
 - 15.2.3. Processus de développement basé sur les cas d'utilisation
 - 15.2.4. Flux de travail fondamentaux du Processus Unifié
- 15.3. La planification dans le contexte du développement Logiciel agile
 - 15.3.1. Caractéristiques du développement Logiciel agile
 - 15.3.2. Différents horizons de planification dans le développement agile
 - 15.3.3. Cadre de développement agile Scrum et planification des horizons temporels
 - 15.3.4. User stories comme unité de planification et d'estimation
 - 15.3.5. Techniques courantes de calcul d'une estimation
 - 15.3.6. Échelles d'interprétation des estimations
 - 15.3.7. Planning poker
 - 15.3.8. Types courants de planification: planification des livraisons et planification des itérations
- 15.4. Styles de conception de Logiciels distribués et architectures Logicielles orientées services
 - 15.4.1. Modèles de communication dans les systèmes Logiciels distribués
 - 15.4.2. Couche intermédiaire
 - 15.4.3. Modèles architecturaux pour les systèmes distribués
 - 15.4.4. Processus général de conception de services Logiciels
 - 15.4.5. Aspects de la conception de services Logiciels
 - 15.4.6. Composition des services
 - 15.4.7. Architecture des services web
 - 15.4.8. Composants d'infrastructure et SOA
- 15.5. Introduction au développement Logiciel guidé par le modèle
 - 15.5.1. Le concept de modèle
 - 15.5.2. Développement Logiciel guidé par le modèle
 - 15.5.3. MDA cadre de développement guidé par le modèle
 - 15.5.4. Éléments d'un modèle de transformation
- 15.6. Conception d'interfaces utilisateur graphiques
 - 15.6.1. Principes de conception d'interfaces utilisateur
 - 15.6.2. Modèles de conception architecturale pour les systèmes interactifs: Modèle Vue Contrôleur Vue (MVC)
 - 15.6.3. Expérience utilisateur (UX User Experience)
 - 15.6.4. Conception centrée sur l'utilisateur
 - 15.6.5. Processus d'analyse et de conception d'une interface utilisateur graphique
 - 15.6.6. Utilisabilité d'interfaces utilisateurs
 - 15.6.7. Accessibilité des interfaces utilisateurs
- 15.7. Conception d'applications web
 - 15.7.1. Caractéristiques des applications web
 - 15.7.2. Interface utilisateur d'une application web
 - 15.7.3. Conception de la navigation
 - 15.7.4. Protocole d'interaction de base pour les applications web
 - 15.7.5. Styles d'architecture pour les applications web
- 15.8. Stratégies et techniques de test des Logiciels et facteurs de qualité des logiciels
 - 15.8.1. Stratégies d'essai
 - 15.8.2. Designs de cas d'essai
 - 15.8.3. Rapport qualité-prix
 - 15.8.4. Modèles de qualité
 - 15.8.5. Famille de normes ISO/IEC 25000 (SQuaRE)
 - 15.8.6. Modèle de qualité du produit (ISO 2501n)
 - 15.8.7. Modèles de qualité des données (ISO 2501n)
 - 15.8.8. Gestion de la qualité des Logiciels
- 15.9. Introduction aux métriques de l'ingénierie logicielle
 - 15.9.1. Concepts de base: Mesures, métriques et indicateurs
 - 15.9.2. Types de métriques en Génie Logiciel
 - 15.9.3. Le processus de mesure
 - 15.9.4. ISO 25024. Mesures externes et de qualité utilisées
 - 15.9.5. Métriques orientées objet

- 15.10. Maintenance et réingénierie des Logiciels
 - 15.10.1. Processus de maintenance
 - 15.10.2. Cadre du processus de maintenance standard. ISO/EIEC 14764
 - 15.10.3. Modèle de processus de réingénierie des Logiciels
 - 15.10.4. Ingénierie inverse

Module 16. Intégration des systèmes

- 16.1. Introduction aux systèmes d'information d'entreprise
 - 16.1.1. Le rôle des systèmes d'information
 - 16.1.2. Conception des systèmes d'information?
 - 16.1.3. Dimensions des systèmes d'information
 - 16.1.4. Processus d'affaires et systèmes d'information
 - 16.1.5. Le département IS/IT
- 16.2. Opportunités et besoins en matière de systèmes d'information dans l'entreprise
 - 16.2.1. Organisations et systèmes d'information
 - 16.2.2. Caractéristiques des organisations
 - 16.2.3. Impact des systèmes d'information sur l'entreprise
 - 16.2.4. Les systèmes d'information pour un avantage concurrentiel
 - 16.2.5. Utilisation de systèmes dans l'administration et la gestion des affaires
- 16.3. Concepts de base des systèmes et technologies de l'information
 - 16.3.1. Données, informations et connaissances
 - 16.3.2. Technologie et systèmes d'information
 - 16.3.3. Composants technologiques
 - 16.3.4. Classification et types de systèmes d'information
 - 16.3.5. Architectures basées sur les services et les processus d'entreprise
 - 16.3.6. Formes d'intégration des systèmes
- 16.4. Systèmes intégrés de gestion des ressources de l'entreprise
 - 16.4.1. Exigences commerciales
 - 16.4.2. Un système d'information intégré pour l'entreprise
 - 16.4.3. Acquisition vs. Développement
 - 16.4.4. Mise en œuvre de l'ERP
 - 16.4.5. Implications pour la gestion
 - 16.4.6. Principaux fournisseurs d'ERP
- 16.5. Systèmes d'information pour la gestion de la chaîne d'approvisionnement et de la relation client
 - 16.5.1. Définition de chaîne d'approvisionnement
 - 16.5.2. Gestion effets de la chaîne d'approvisionnement
 - 16.5.3. Le rôle des systèmes d'information
 - 16.5.4. Solutions pour la gestion de chaîne d'approvisionnement
 - 16.5.5. La gestion des relations avec les clients
 - 16.5.6. Le rôle des systèmes d'information
 - 16.5.7. Mise en œuvre d'un système CRM
 - 16.5.8. Facteurs critiques de succès dans la mise en œuvre de la CRM
 - 16.5.9. CRM, e-CRM et autres tendances
- 16.6. Prise de décision en matière d'investissement dans les TIC et planification des systèmes d'information
 - 16.6.1. Critères de décision en matière d'investissement dans les TIC
 - 16.6.2. Lier le projet au plan de gestion et d'affaires
 - 16.6.3. Implications en termes de gestion
 - 16.6.4. Refonte des processus d'affaires
 - 16.6.5. Décision de la direction sur les méthodologies de mise en œuvre
 - 16.6.6. Nécessité d'une planification des systèmes d'information
 - 16.6.7. Objectifs, participants et calendrier
 - 16.6.8. Structure et développement du plan des systèmes
 - 16.6.9. Suivi et mise à jour
- 16.7. Considérations de sécurité dans l'utilisation des TIC
 - 16.7.1. Analyse des risques
 - 16.7.2. Sécurité des systèmes d'information
 - 16.7.3. Conseils pratiques
- 16.8. Faisabilité de la mise en œuvre de projets TIC et aspects financiers dans les projets de systèmes d'information
 - 16.8.1. Description et objectifs
 - 16.8.2. Participants au SVE
 - 16.8.3. Techniques et pratiques
 - 16.8.4. Structure des coûts
 - 16.8.5. Projection financière
 - 16.8.6. Les présupposés

- 16.9. Business Intelligence
 - 16.9.1. Qu'est-ce que la Business Intelligence?
 - 16.9.2. Stratégie et mise en œuvre de la BI
 - 16.9.3. Le présent et l'avenir de la BI
- 16.10. ISO/IEC 12207
 - 16.10.1. Qu'est-ce que "ISO/IEC 12207"?
 - 16.10.2. Analyse des systèmes d'informations
 - 16.10.3. Conception d'un système d'information
 - 16.10.4. Mise en œuvre et acceptation du système d'information

Module 17. Réutilisation de Software

- 17.1. Présentation de réutilisation Software
 - 17.1.1. Qu'est-ce que la la réutilisation des Logiciels
 - 17.1.2. Avantages et inconvénients de la réutilisation des Logiciels
 - 17.1.3. Principales techniques de réutilisation des Logiciels
- 17.2. Introduction aux modèles de conception
 - 17.2.1. Qu'est-ce qu'un modèle de conception?
 - 17.2.2. Catalogue des principaux modèles de conception
 - 17.2.3. Comment utiliser les modèles pour résoudre les problèmes de conception
 - 17.2.4. Comment choisir le meilleur modèle de conception
- 17.3. Modèles de création
 - 17.3.1. Modèles de création
 - 17.3.2. Modèle de fabrique abstraite
 - 17.3.3. Exemple de mise en œuvre du modèle de fabrique abstraite
 - 17.3.4. Modèle de construction
 - 17.3.5. Exemple de mise en œuvre du constructeur
 - 17.3.6. Usine abstraite vs. Builder
- 17.4. Les modèles de création(II)
 - 17.4.1. Patron Factory Method
 - 17.4.2. Factory Method vs. Abstract Factory
 - 17.4.3. Modèle Singleton
- 17.5. Modèles structurels
 - 17.5.1. Modèles structurels
 - 17.5.2. Modèle Adapter
 - 17.5.3. Modèle Bridge
- 17.6. Les modèles structurelles (II)
 - 17.6.1. Modèle Composite
 - 17.6.2. Modèle Decorador
- 17.7. Les modèles structurelles (III)
 - 17.7.1. Modèle Facade
 - 17.7.2. Modèle Proxy
- 17.8. Modèles de comportement
 - 17.8.1. Concept de modèles comportementaux
 - 17.8.2. Modèle de comportement: Chaîne de responsabilité
 - 17.8.3. Modèle de comportement Ordre
- 17.9. Modèles de comportement (II)
 - 17.9.1. Modèle d'Interprète ou Interpreter
 - 17.9.2. Modèle Iterador
 - 17.9.3. Modèle Observador
 - 17.9.4. Modèle de Stratégie
- 17.10. Frameworks
 - 17.10.1. Concept de framework
 - 17.10.2. Développement à l'aide de frameworks
 - 17.10.3. Model View Controller
 - 17.10.4. Frameworks pour la conception d'interfaces utilisateur graphiques
 - 17.10.5. Frameworks pour le développement d'applications web
 - 17.10.6. Frameworks pour gérer la persistance des objets dans les bases de données

Module 18. Services informatiques

- 18.1. La transformation digitale (I)
 - 18.1.1. L'innovation dans l'entreprise
 - 18.1.2. La gestion de la production
 - 18.1.3. La gestion financière
- 18.2. La transformation digitale (II)
 - 18.2.1. Le marketing
 - 18.2.2. Gestion des R H
 - 18.2.3. Un système d'information complet
- 18.3. Étude de cas
 - 18.3.1. Présentation de l'entreprise
 - 18.3.2. Méthodologie pour analyser l'acquisition de TI
 - 18.3.3. Détermination des coûts, bénéfices et risques
 - 18.3.4. Évaluation économique de l'investissement
- 18.4. la gouvernance et la gestion des TIC
 - 18.4.1. Définition de la gouvernance des technologies et des systèmes d'information
 - 18.4.2. Différence entre la gouvernance et la gestion de l'IST
 - 18.4.3. Cadres pour la gouvernance et la gestion des TSI
 - 18.4.4. Normes, gouvernance et gestion des TSI
- 18.5. Gouvernance d'entreprise dans le domaine des TIC
 - 18.5.1. Qu'est-ce qu'une bonne gouvernance d'entreprise?
 - 18.5.2. Contexte de la gouvernance des TIC
 - 18.5.3. Norme ISO/IEC 38500:2008
 - 18.5.4. Mise en œuvre d'une bonne gouvernance des TIC
 - 18.5.5. Gouvernance et meilleures pratiques en matière de TIC
 - 18.5.6. La gouvernance d'entreprise. Vue d'ensemble et tendances
- 18.6. Objectifs de contrôle pour les technologies de l'information et les technologies connexes (COBIT)
 - 18.6.1. Cadre de mise en œuvre
 - 18.6.2. Domaine: Planification et organisation
 - 18.6.3. Domaine: Acquisition et mise en œuvre
 - 18.6.4. Domaine: Livraison et soutien
 - 18.6.5. Domaine: Contrôle et évaluation
 - 18.6.6. Application du Guide COBIT
- 18.7. La bibliothèque d'infrastructure des technologies de l'information (ITIL)
 - 18.7.1. Introduction à ITIL
 - 18.7.2. Stratégies de service
 - 18.7.3. Conception des services
 - 18.7.4. Transition des services
 - 18.7.5. Opération des services
 - 18.7.6. Amélioration des services
- 18.8. Système de gestion des services
 - 18.8.1. Principes de base de la norme UNE-ISO/IEC 20000-1
 - 18.8.2. La structure de la série de normes ISO/CEI 20000
 - 18.8.3. Exigences du système de gestion des services
 - 18.8.4. Conception et transition de services nouveaux ou modifiés
 - 18.8.5. Processus de prestation de services
 - 18.8.6. Groupes de processus
- 18.9. Système de gestion des actifs des Software
 - 18.9.1. Justification du besoin
 - 18.9.2. Antécédents
 - 18.9.3. Présentation de 19770
 - 18.9.4. Mise en œuvre de la gestion
- 18.10. Gestion de la continuité des activités
 - 18.10.1. Plan de continuité des activités
 - 18.10.2. Mise en place du BCM



“

Tirez parti de vos connaissances antérieures et demandez la validation des matières que vous avez déjà réussies, optimisant ainsi votre expérience dans ce programme”

04

Objectifs pédagogiques

Ce programme en Ingénierie du Logiciel a été conçu pour fournir aux professionnels de l'informatique les outils avancés nécessaires pour propulser leur carrière dans le développement de Logiciels modernes, adaptés à l'évolution des demandes du marché. Grâce à des connaissances techniques spécialisées, les étudiants augmenteront considérablement leurs possibilités d'évolution professionnelle, ce qui leur ouvrira les portes de postes clés dans des entreprises de premier plan du secteur.



```
if _operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
if _operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
#selection at the end -add back the deselected  
mirror_ob.select= 1  
modifier_ob.select=1  
...context.scene.objects.active = modifier_ob  
print("Selected" + str(modifier_ob.name))  
#mirror_ob.select = 0  
... = bpy.context.selected_objects  
...data.objects[0].name)
```



*Un programme conçu pour devenir
l'ultime coup de pouce vers une carrière
réussie dans l'Ingénierie du Logiciel"*

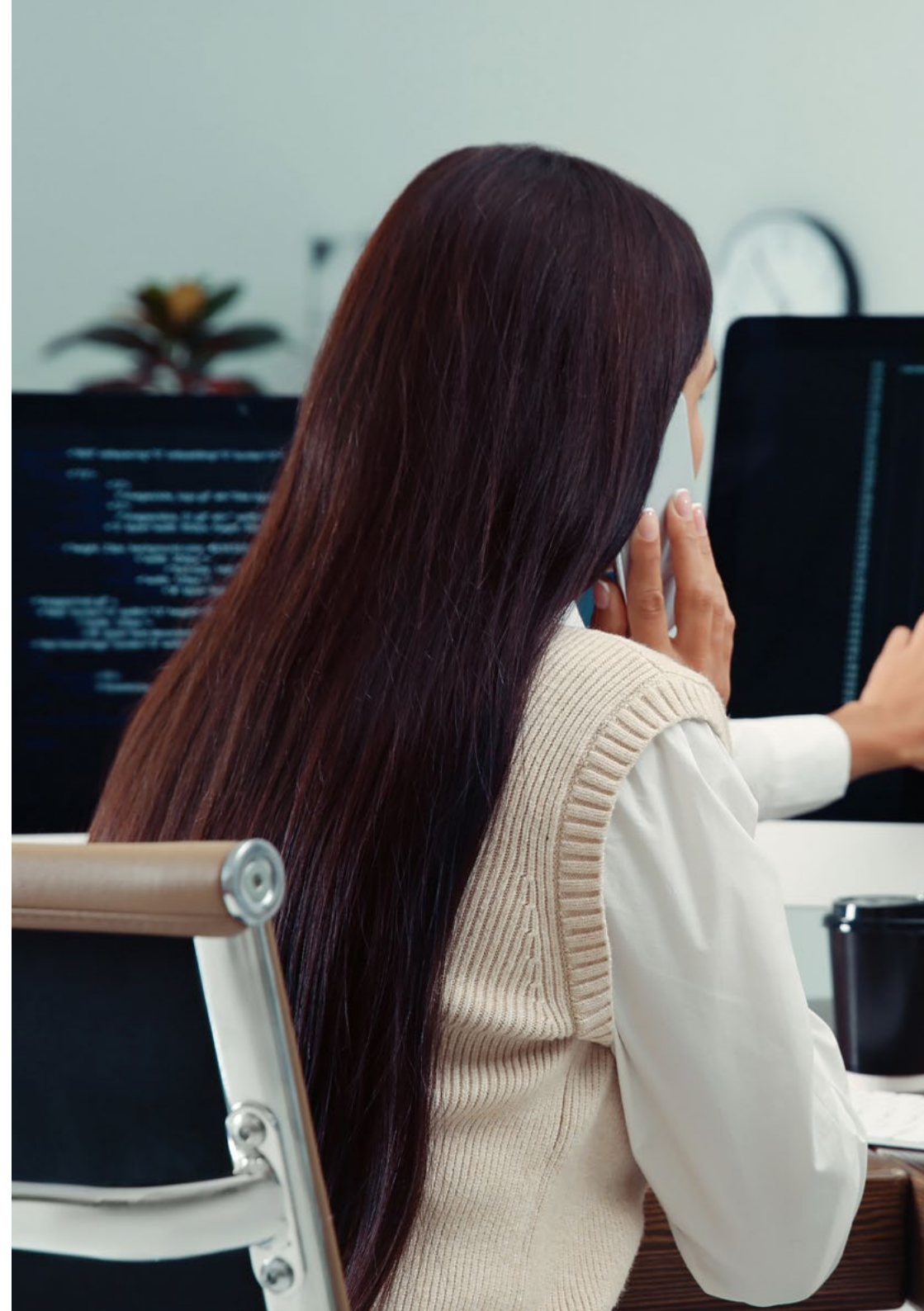


Objectifs généraux

- ♦ Acquérir de nouvelles compétences nécessaires et exigées en termes de nouvelles technologies et des derniers développements en matière de Logiciels
- ♦ Compléter les connaissances acquises par des compétences dans le domaine du calcul et de la structure informatique, y compris les bases mathématiques, statistiques et physiques essentielles à l'Ingénierie
- ♦ Élargir les connaissances en Ingénierie du Logiciel et en Systèmes Informatiques avec les derniers développements et la méthodologie la plus innovante
- ♦ Aborder des projets et des environnements Logiciels complexes, en sachant comment fournir des solutions intelligentes à divers problèmes

“

Boostez votre carrière avec un programme de spécialisation conçu pour prendre la direction stratégique de l'Ingénierie du Logiciel”





Objectifs spécifiques

Module 1. Méthodologies, développement et qualité du Génie Logiciel

- ♦ Former à l'utilisation des méthodologies agiles et traditionnelles dans le développement de Logiciels
- ♦ Assurer la qualité du produit final par l'application de normes et de bonnes pratiques à chaque phase du cycle de vie du Logiciel

Module 2. Gestion de projets de Software

- ♦ Développer les compétences nécessaires pour planifier, exécuter et gérer des projets Logiciels
- ♦ Appliquer les méthodologies de gestion de projet et les outils de planification pour assurer une livraison dans les délais et le budget

Module 3. Plateformes de développement du Logiciel

- ♦ Fournir des connaissances sur les différentes plates-formes de développement de Logiciels
- ♦ Aborder l'utilisation d'environnements de développement, de *frameworks* et d'outils qui optimisent la création d'applications robustes et évolutives

Module 4. Informatique client web

- ♦ Se former au développement d'applications web côté client, en utilisant des technologies telles que HTML, CSS et JavaScript
- ♦ Créer des interfaces interactives optimisées pour l'expérience utilisateur

Module 5. Informatique des serveurs web

- ♦ Développer les compétences nécessaires pour gérer et développer des applications web côté serveur
- ♦ Manipuler des technologies telles que PHP, Node.js ou Python, et gérer des bases de données et des services backend



Module 6. Gestion de la sécurité

- ♦ Fournir les connaissances et les compétences nécessaires pour gérer la sécurité dans les environnements de développement de Logiciels
- ♦ Appliquer des politiques et des stratégies pour la protection des données, l'accès et la sécurité de l'infrastructure technologique

Module 7. Sécurité des systèmes d'information

- ♦ Fournir une compréhension approfondie de la sécurité dans les systèmes d'information
- ♦ Analyser la protection des réseaux et des serveurs jusqu'à la mise en œuvre de protocoles et de normes de sécurité dans les infrastructures technologiques

Module 8. Sécurité du Software

- ♦ Former au développement de Logiciels sécurisés
- ♦ Identifier, atténuer et prévenir les vulnérabilités dans le code et les processus de développement des applications

Module 9. Qualité et audit des systèmes d'information

- ♦ Développer les compétences pour réaliser des audits de qualité des systèmes d'information
- ♦ S'assurer de l'efficacité des contrôles de sécurité, de la conformité aux réglementations et de l'optimisation des processus technologiques

Module 10. Administration des serveurs web

- ♦ Former à la gestion et à l'administration des serveurs web
- ♦ Approfondir la maintenance des serveurs qui hébergent des applications et des services en ligne

Module 11. Sécurité des applications online

- ♦ Fournir les connaissances nécessaires pour sécuriser les applications en ligne
- ♦ Aborder des aspects tels que l'authentification, la protection des données, la gestion des accès et la prévention des attaques dans les environnements web

Module 12. Ingénierie Software

- ♦ Fournir une compréhension complète des principes et des pratiques du génie Logiciel
- ♦ Concevoir, développer et maintenir des applications logicielles efficaces, évolutives et de haute qualité

Module 13. Génie Logiciel avancée

- ♦ Approfondir les techniques avancées d'Ingénierie du Logiciel
- ♦ Approfondir les méthodologies de développement sophistiquées, les modèles de conception, l'architecture Logicielle et l'optimisation des processus pour les projets complexes

Module 14. Ingénierie des exigences

- ♦ Développer des compétences dans l'identification, l'analyse et la documentation des exigences Logicielles
- ♦ Veiller à ce que les produits finis répondent aux besoins et aux attentes des utilisateurs



Module 15. Processus d'Ingénierie Software

- ♦ Former aux processus fondamentaux de l'ingénierie Logicielle, de la planification et de la conception à la mise en œuvre, aux tests et à la maintenance
- ♦ Améliorer la qualité et l'efficacité du cycle de vie des Logiciels

Module 16. Intégration des systèmes

- ♦ Intégrer différents systèmes Logiciels, bases de données et applications
- ♦ Créer des solutions technologiques cohérentes et efficaces pour résoudre des problèmes commerciaux complexes

Module 17. Réutilisation de Software

- ♦ Appliquer des techniques et des principes pour la réutilisation de composants et de modules logiciels dans de nouveaux projets
- ♦ Optimiser le temps et les coûts de développement en créant des bibliothèques ou des *frameworks* réutilisables

Module 18. Services informatiques

- ♦ Concevoir, mettre en œuvre et gérer des services informatiques
- ♦ Améliorer l'infrastructure technologique pour assurer la continuité des activités et l'efficacité opérationnelle

05

Opportunités de carrière

À l'issue de ce programme universitaire, les professionnels auront une solide compréhension des techniques, des méthodologies et des outils nécessaires pour mener à bien des projets de développement de Logiciels modernes adaptés aux besoins actuels du marché. Les diplômés seront capables de concevoir, de mettre en œuvre et de gérer des solutions technologiques avancées, en garantissant la qualité, la sécurité et l'efficacité dans toutes les phases du développement. Ils seront également capables de diriger des équipes pluridisciplinaires, de collaborer dans des environnements agiles et de prendre des décisions stratégiques fondées sur des normes internationales.



“

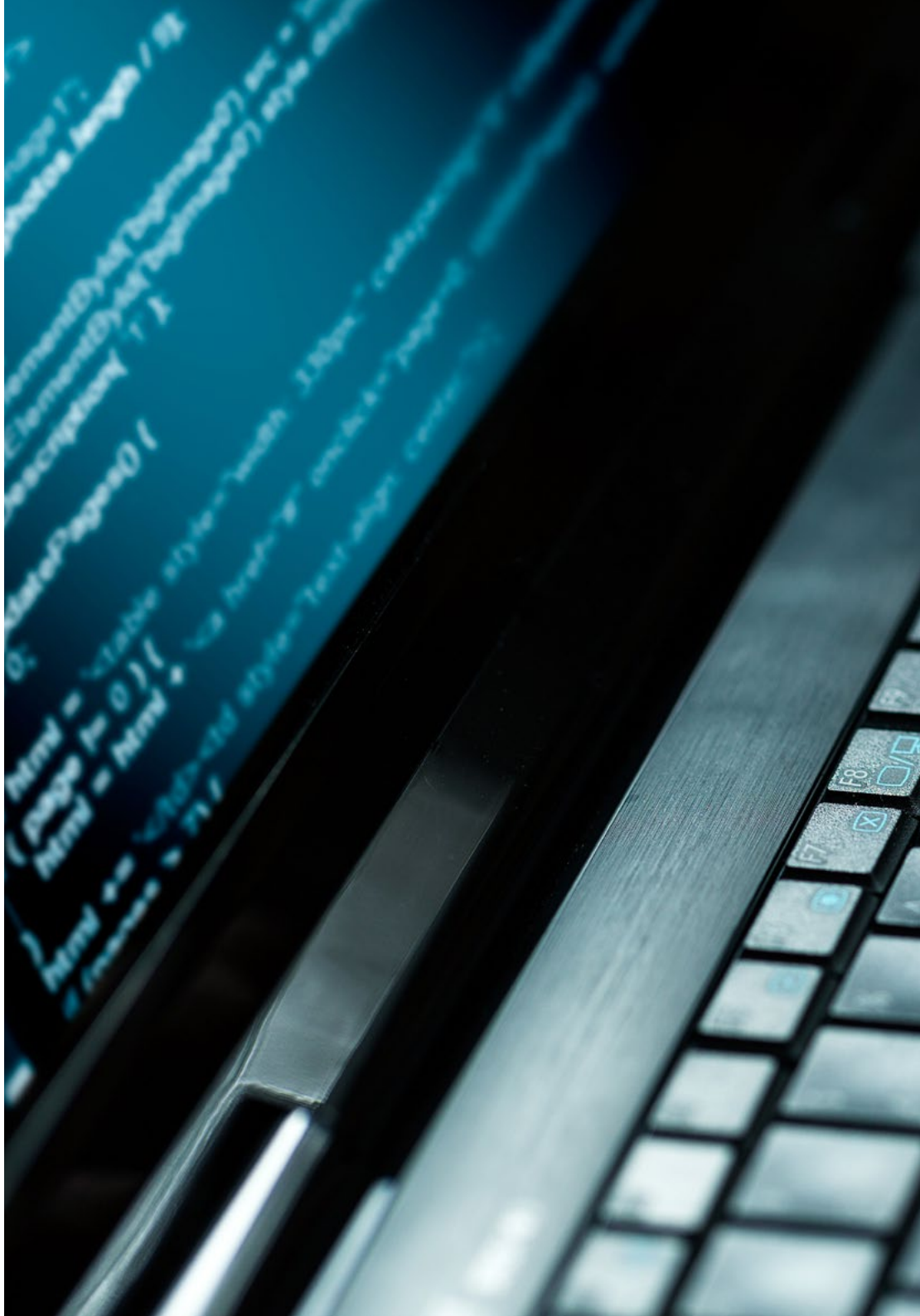
Vous appliquerez des approches innovantes et des solutions efficaces qui optimisent le développement et la gestion de Logiciels dans le cadre de projets technologiques complexes”


Profil des diplômés

Les diplômés du programme en Ingénierie du Logiciel sont hautement qualifiés pour faire face aux défis technologiques du développement moderne de Logiciels. En outre, il/elle possède une connaissance approfondie des méthodologies avancées, des outils de développement et des normes internationales, nécessaires pour créer des solutions innovantes et efficaces. Vous êtes également prêt à concevoir, mettre en œuvre et gérer des projets technologiques à fort impact, à collaborer avec des équipes pluridisciplinaires et à mener des initiatives stratégiques dans des environnements dynamiques et concurrentiels, en garantissant la qualité, la sécurité et l'évolutivité des applications développées.

Vous combinerez des connaissances théoriques avec des compétences pratiques en matière de conception de Logiciels et de méthodologies de développement avancées.

- ♦ **Conception et développement de Logiciels:** Capacité à concevoir et à mettre en œuvre des applications avancées, en intégrant des modèles de conception, la modélisation UML et des plateformes de développement telles que Java et .NET.
- ♦ **Gestion de projets technologiques:** Capacité à planifier, superviser et diriger des projets technologiques en utilisant des méthodologies agiles telles que Scrum et en appliquant les normes internationales ISO/IEC
- ♦ **Sécurité dans le développement:** Compréhension approfondie des meilleures pratiques en matière de sécurité informatique, de gestion des risques et de protection des données à tous les stades du cycle de vie des Logiciels
- ♦ **Déploiement dans des environnements complexes:** Capacité à développer des applications mobiles, web et basées sur le cloud en utilisant des technologies telles que REST, SOA et MVC
- ♦ **Résolution de problèmes:** Application de la pensée critique et de stratégies innovantes pour identifier et résoudre les défis technologiques dans les projets Logiciels
- ♦ **Compétence numérique:** Maîtrise des outils technologiques avancés et des plateformes numériques pour le développement, le test et la gestion des systèmes





Après avoir obtenu le titre de Mastère Avancé, vous serez en mesure d'utiliser vos connaissances et vos compétences dans les postes suivants:

- 1. Directeur de projet de Software:** Responsable de la planification et de l'exécution de projets complexes, en veillant au respect des objectifs de qualité, de délai et de budget.
- 2. Architecte Logiciel:** Spécialiste de la conception et de la construction de systèmes complexes et évolutifs.
- 3. Ingénieur en sécurité informatique:** Responsable de la protection des systèmes, des applications et des données contre les vulnérabilités et les attaques.
- 4. Développeur full-stack:** Professionnel capable de créer des solutions complètes tant du côté client que du côté serveur.
- 5. Consultant en technologie:** Conseiller expert dans la mise en œuvre de technologies avancées et de stratégies de transformation numérique.
- 6. Spécialiste des applications en nuage:** Concepteur et implémenteur de solutions basées sur SaaS, PaaS et IaaS.
- 7. Administrateur de systèmes:** Gestionnaire de serveurs, de réseaux et de plateformes technologiques, optimisant leurs performances et leur sécurité.
- 8. Analyste de la qualité de Software:** Évaluateur de la fonctionnalité, de la convivialité et de la performance des applications technologiques.
- 9. Consultant en ingénierie des exigences:** Spécialiste de l'analyse, de la documentation et de la validation des exigences dans les projets de Software.



En suivant ce programme, vous accéderez à des postes clés dans l'industrie technologique et vous vous distinguerez en tant que leader dans le domaine de l'Ingénierie du Logiciel"

06

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”

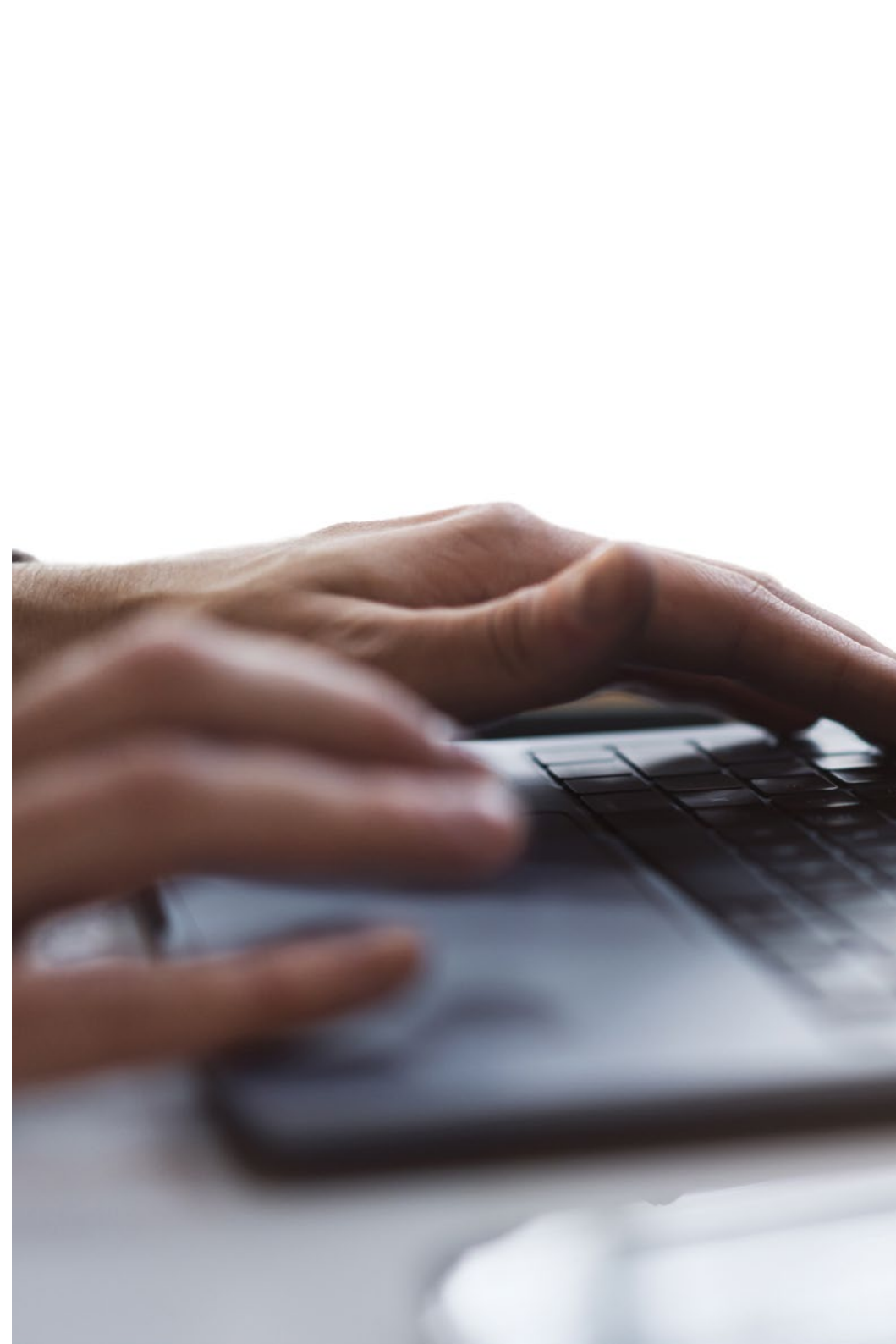
L'étudiant: la priorité de tous les programmes de TECH

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

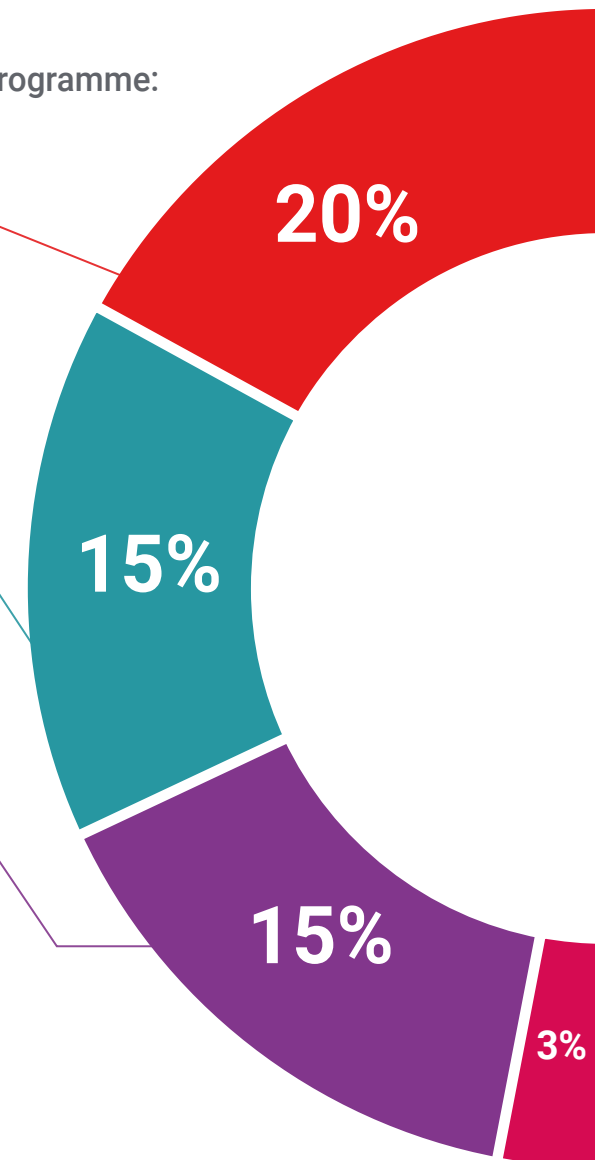
Nous présentons les contenus de manière attrayante et dynamique dans des dossiers multimédias qui incluent de l'audio, des vidéos, des images, des diagrammes et des cartes conceptuelles afin de consolider les connaissances.

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.



07

Corps Enseignant

Les enseignants de ce programme en Ingénierie du Logiciel se distinguent par leurs connaissances techniques approfondies et leur expérience pratique dans l'industrie du Logiciel. En fait, ils ont dirigé d'importants projets dans des entreprises de premier plan du secteur, ce qui leur permet de partager non seulement des théories avancées, mais aussi des applications pratiques et des cas réels qui enrichiront la préparation des diplômés. En outre, leur engagement en faveur de l'innovation et de la recherche les pousse à se tenir au courant des dernières tendances et méthodologies de développement.

“

Apprendre auprès de professeurs expérimentés et compétents en Informatique et en Ingénierie du Logiciel”

Directeur Invité International

Darren Pulsipher est un **architecte logiciel** très expérimenté, un innovateur avec une expérience internationale distinguée dans le **développement de software et de firmware**. En fait, il possède des compétences très développées en matière de **communication, de gestion de projets et d'affaires**, qui lui ont permis de diriger d'importantes initiatives au niveau mondial.

Il a également occupé des postes à haute responsabilité tout au long de sa carrière, comme celui d'**Architecte en Chef des Solutions pour le Secteur Public** chez Intel Corporation, où il a promu des **activités modernes, des processus et des technologies** pour les clients, les partenaires et les utilisateurs du **secteur public**. En outre, il a fondé Yoly Inc., dont il a également été le **PDG**, afin de développer un **outil d'agrégation et de diagnostic des réseaux sociaux** basé sur un **Software AS A Service (SaaS)**, utilisant les technologies du **Big Data** et du **Web 2.0**.

Il a également travaillé dans d'autres entreprises, en tant que **Directeur Senior de l'Ingénierie**, chez Dell Technologies, où il a dirigé l'**Unité Commerciale de Big Data dans le Cloud**. Il a dirigé des équipes aux **Etats-Unis** et en **Chine** dans la gestion de grands projets et a restructuré des divisions commerciales pour une intégration réussie. Il a également travaillé en tant que **Directeur de la Technologie de l'Information (Chief Information Officer)** chez XanGo, où il a géré des projets tels que le **support du Help Desk**, le **support de production** et le **développement de solutions**.

Parmi les nombreuses spécialités dans lesquelles il est expert figurent la technologie **Edge to Cloud**, la **cybersécurité**, l'**Intelligence Artificielle Générative**, le **développement de logiciels**, la **technologie de mise en réseau**, le **développement cloud-native** et l'**écosystème des conteneurs**. Des connaissances qu'il a partagées grâce au **podcast** et à la **lettre d'information hebdomadaire** "**Embracing Digital Transformation**", qu'il a produits et présentés, et qui aident les organisations à réussir leur **transformation numérique** en s'appuyant sur les **personnes, les processus et la technologie**.



M. Pulsipher, Darren

- Architecte en Chef des Solutions pour le Secteur Public chez Intel, Californie, États-Unis
- Présentateur et Producteur de "*Embracing Digital Transformation*", Californie
- Fondateur et PDG de Yoly Inc, Arkansas
- Directeur Senior de l'Ingénierie chez Dell Technologies, Arkansas
- Directeur des Technologies de l'Information (*Chief Information Officer*) chez XanGo, Utah
- Architecte Senior chez Cadence Design Systems, Californie
- Gestionnaire Principal des Processus de Projet chez Lucent Technologies, Californie
- Ingénieur Logiciel chez Cemax-Icon, Californie
- Ingénieur Logiciel à ISG Technologies, Canada
- MBA en Gestion des Technologies de l'Université de Phoenix
- Licence en Sciences Informatiques et Ingénierie Électrique de l'Université Brigham Young

“

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

08 Diplôme

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



“

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

Ce **Mastère Avancé en Ingénierie du Logiciel** contient le programme le plus complet et le plus actualisé du marché.

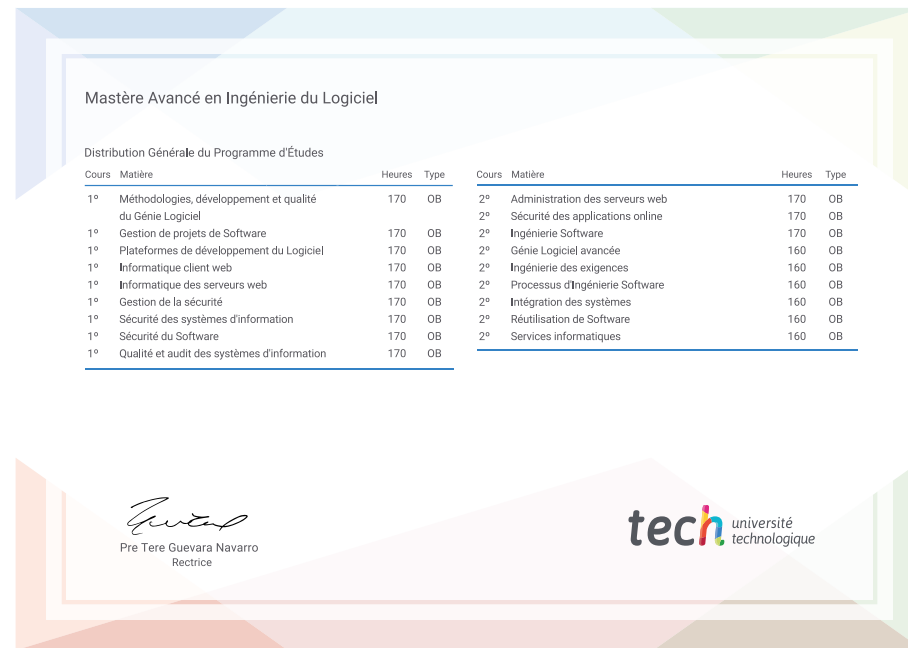
Après avoir passé l'évaluation, l'étudiant recevra par courrier* avec accusé de réception son diplôme de **Mastère Avancé** délivrée par **TECH Université Technologique**.

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

Diplôme: **Mastère Avancé en Ingénierie du Logiciel**

Modalité: **en ligne**

Durée: **2 ans**



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

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

tech université
technologique

Mastère Avancé Ingénierie du Logiciel

- » Modalité: en ligne
- » Durée: 2 ans
- » Qualification: TECH Université Technologique
- » Horaire: à votre rythme
- » Examens: en ligne

Mastère Avancé Ingénierie du Logiciel

