

Mastère Spécialisé

Création des Interfaces et Applications en Réseau



Mastère Spécialisé Création des Interfaces et Applications en Réseau

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

Accès au site web: www.techtitute.com/fr/informatique/master/master-creation-interfaces-applications-reseau

Sommaire

01

Présentation

page 4

02

Objectifs

page 8

03

Compétences

page 14

04

Structure et contenu

page 18

05

Méthodologie

page 32

06

Diplôme

page 40

01

Présentation

Pour être compétitif dans le domaine de l'informatique, il faut avoir une solide connaissance des développements et des mises à jour les plus pertinents en matière de génie logiciel, de fondements mathématiques, de statistiques et d'autres domaines. Dans ce programme, nous avons construit un parcours intensif de spécialisation à travers lequel le professionnel sera en mesure de se former scientifiquement et technologiquement dans l'ingénierie informatique, avec qualité, sécurité et un objectif ultime d'excellence.



“

Notre concept innovant de télépratique vous donnera l'opportunité d'apprendre à travers un programme intensif et immersif, et vous permettra d'apprendre plus rapidement et d'avoir une vision plus réaliste des contenus: "Learning from an Expert"

Cette formation développera les concepts nécessaires pour travailler dans la création d'interfaces, avec la garantie de la maîtrise de tous les domaines de connaissance transversaux à ce sujet. Tout au long du programme, des approches didactiques innovantes seront proposées afin d'acquérir une connaissance approfondie de l'architecture d'une application distribuée. Les aspects particuliers de l'architecture client-serveur seront présentés et les fondamentaux et les développements essentiels des langages de programmation les plus couramment utilisés seront abordés, la différenciation entre les langages parmi beaucoup d'autres questions fondamentales pour le professionnel.

Cette connaissance essentielle devient également la première étape pour accéder à la capacité de développement de ce type de technologie.

Tout au long de cette formation, un scénario de travail réel est fourni afin de pouvoir évaluer la pertinence de son application dans le projet, en évaluant les indications réelles, le mode de développement et les attentes que l'on peut avoir en ce qui concerne les résultats.

En acquérant de l'expérience, vous apprendrez à développer les connaissances dont vous avez besoin pour progresser dans ce domaine de travail. Cette spécialisation, qui exige nécessairement une expérience, parvient à conjuguer l'apprentissage à distance et l'enseignement pratique, offrant ainsi une option unique pour donner à votre CV le coup de pouce que vous attendez.



Devenez l'un des professionnels les plus prisés du moment: formez-vous à l'ingénierie informatique grâce au Mastère Spécialisé le plus complet et le plus actualisé en Création des Interfaces et Applications en Réseau"

Ce **Mastère Spécialisé en Création des Interfaces et Applications en Réseau** contient le programme académique le plus complet et le plus actuel du marché. Les principales caractéristiques sont les suivantes:

- ◆ Les dernières technologies en matière de logiciels d'enseignement en ligne
- ◆ Un système d'enseignement intensément visuel, soutenu par un contenu graphique et schématique facilitant la compréhension et l'assimilation
- ◆ Le développement de cas pratiques présentés par des experts en activité
- ◆ Systèmes de vidéos interactives de pointe
- ◆ Enseignement basé sur la télépratique
- ◆ Des systèmes d'améliorations et de mises à jour continues
- ◆ Un apprentissage autorégulable: compatibilité totale avec d'autres occupations
- ◆ Les exercices pratiques d'auto-évaluation et de suivi de la progression de l'apprentissage
- ◆ Groupes de soutien et interactions pédagogiques: questions à l'expert, forums de discussion et d'échange de connaissances
- ◆ La communication avec l'enseignant et le travail de réflexion individuel
- ◆ Les contenus sont disponibles à partir de tout dispositif fixe ou portable et doté d'une connexion internet
- ◆ Banques de documentation complémentaire disponibles en permanence, même après le programme

“ Avec une conception méthodologique basée sur des techniques d'enseignement variées, ce Mastère Spécialisé en Création des Interfaces et Applications en Réseau vous offrira les différentes approches pédagogiques pour vous permettre d'apprendre de manière dynamique et efficace”

Le corps enseignant est composé de professionnels issus de différents domaines liés à cette spécialité. De cette manière, TECH s'assure qu'il fournit l'objectif de mise à jour éducative recherché. Une équipe multidisciplinaire de professionnels formés et expérimentés dans différents environnements, qui développeront les connaissances théoriques de manière efficace. Ils mettront également au service du cours, les connaissances pratiques issues de leur propre expérience: une des qualités différentielles de cette formation.

Cette maîtrise du sujet est complétée par l'efficacité de la conception méthodologique. Conçu par une équipe pluridisciplinaire d'experts en *e-learning* qui intègre les dernières avancées en matière de technologie éducative. Ainsi, vous pourrez étudier avec une gamme d'outils multimédias pratiques et polyvalents qui vous apporteront la fonctionnalité dont vous avez besoin dans votre formation.

La conception de ce programme est basée sur l'Apprentissage par Problèmes: une approche qui conçoit l'apprentissage comme un processus éminemment pratique. Pour y parvenir à distance, nous utiliserons la télépratique: à l'aide d'un innovant système de vidéos interactives, et le Learning from an Expert vous acquérez les connaissances comme si vous y étiez confronté. Un concept qui permet d'intégrer et de fixer votre apprentissage de manière plus réaliste et permanente.

Un Mastère Spécialisé qui vous permettra de travailler dans tous les domaines de la Création des Interfaces et Applications en Réseau avec la solvabilité d'un professionnel de haut niveau.

Grâce à l'expérience de professionnels en exercice qui vous offriront une connaissance réelle, immédiate et concrète de ce domaine d'activité.



02 Objectifs

L'objectif de TECH est de fournir aux professionnels des informations actualisées sur la création d'interfaces et d'applications de réseau. Cela permet à l'ingénieur en systèmes d'acquérir de nouvelles compétences pour la réalisation de programmes informatiques plus complets. C'est un objectif réalisable en quelques mois grâce à un programme d'études axé sur les besoins actuels du secteur.



“

*Développez vos compétences
en informatique et en software
et préparez-vous à concourir
parmi les meilleurs de l'industrie"*



Objectifs généraux

- ◆ Préparer scientifiquement et technologiquement à la pratique de l'ingénierie informatique
- ◆ Obtenir des connaissances complètes dans le domaine de l'informatique
- ◆ Obtenir des connaissances complètes dans le domaine de la structure informatique
- ◆ Acquérir les connaissances nécessaires en matière de génie logiciel
- ◆ Passer en revue les fondements mathématiques, statistiques et physiques essentiels à cette discipline

“

Une voie vers la formation et la croissance professionnelle qui vous propulsera vers une grande compétitivité sur le marché du travail”





Objectifs spécifiques

Module 1. Interaction homme-ordinateur

- ◆ Acquérir des connaissances solides en matière d'interaction homme-machine et de création d'interfaces utilisables
- ◆ Comprendre l'importance de l'utilisabilité des applications et pourquoi elle doivent être prises en compte lors de la conception de nos logiciels
- ◆ Comprendre les différents types de diversité humaine, les limites qu'ils impliquent et comment adapter les interfaces en fonction des besoins spécifiques de chacun d'entre eux
- ◆ Apprendre le processus de conception d'une interface, de l'analyse des besoins à l'évaluation, en passant par les différentes étapes intermédiaires nécessaires à la création d'une interface adaptée
- ◆ Connaître les différentes lignes directrices en matière d'accessibilité, les normes qui les établissent et les outils qui permettent de les évaluer
- ◆ Comprendre les différentes méthodes d'interaction avec l'ordinateur, à l'aide de périphériques et de dispositifs

Module 2. Bases de données

- ◆ Apprendre les différentes applications et finalités des systèmes de base de données, ainsi que leur fonctionnement et leur architecture
- ◆ Comprendre le modèle relationnel, depuis sa structure et ses opérations jusqu'à l'algèbre relationnelle étendue
- ◆ Apprendre en profondeur ce que sont les bases de données SQL, comment elles fonctionnent, la définition des données et la création de requêtes, des plus basiques aux plus avancées

- ♦ Apprendre à concevoir des bases de données en utilisant le modèle entité-relationnel, comment créer des diagrammes et les caractéristiques du modèle E-R étendu
- ♦ Approfondir la conception des bases de données relationnelles, en analysant les différentes formes normales et les algorithmes de décomposition
- ♦ Poser les bases pour comprendre le fonctionnement des bases de données NoSQL, et présenter la base de données Mongo DB

Module 3. Développement d'applications en réseau

- ♦ Apprendre les caractéristiques du langage de balisage HTML et son utilisation dans la création de sites web avec les feuilles de style CSS
- ♦ Apprendre à utiliser le langage de programmation orienté navigateur JavaScript, et certaines de ses principales caractéristiques
- ♦ Comprendre les concepts de la programmation orientée composants et de l'architecture des composants
- ♦ Apprendre à utiliser le *Framework* para *Front-End Bootstrap* pour la conception de sites web
- ♦ Comprendre la structure du modèle contrôleur-vue dans le développement de sites web dynamiques
- ♦ Apprendre l'architecture orientée service et les bases du protocole HTTP

Module 4. Logiciels Libres et la Connaissance Ouverte

- ♦ Apprendre les concepts de logiciel libre et de connaissance ouverte, ainsi que les différents types de licences associées
- ♦ Connaître les principaux outils gratuits disponibles dans différents domaines tels que les systèmes d'exploitation, la gestion d'entreprise, les gestionnaires de contenu et la création de contenu multimédia
- ♦ Comprendre l'importance et les avantages des logiciels libres dans le monde de l'entreprise, tant du point de vue de leurs caractéristiques que de leurs coûts
- ♦ Approfondir la connaissance du système d'exploitation GNU/Linux, ainsi que des différentes distributions existantes, et comment réaliser des adaptations personnalisées

- ♦ Apprendre le fonctionnement et le développement de WordPress, sachant que ce CMS représente plus de 35% des sites web actifs dans le monde, et plus de 60% dans le cas des CMS
- ♦ Comprendre le fonctionnement du système d'exploitation des appareils mobiles Android, ainsi que les bases du développement d'applications mobiles en mode natif et avec des *Frameworks* multiplateformes.

Module 5. Bases de Données Avancées

- ♦ Introduire les différents systèmes de bases de données actuellement disponibles sur le marché
- ♦ Apprendre l'utilisation de XML et des bases de données pour le web
- ♦ Comprendre le fonctionnement des bases de données avancées telles que les bases de données parallèles et distribuées
- ♦ Comprendre l'importance de l'indexation et de l'association dans les systèmes de bases de données
- ♦ Comprendre le fonctionnement des systèmes de traitement et de recherche transactionnels
- ♦ Acquérir des connaissances relatives aux bases de données non relationnelles et à l'exploration de données

Module 6. Génie Logiciel

- ♦ Connaître le cadre de référence du génie logiciel et la norme ISO/IEC 12207
- ♦ Apprendre les caractéristiques du processus de développement logiciel unifié et de la planification dans le contexte du développement logiciel agile
- ♦ Connaître les différents styles de conception de logiciels distribués et d'architectures logicielles orientées aux services
- ♦ Apprendre les concepts essentiels d'une interface graphique de l'utilisateur
- ♦ Comprendre les bases du développement d'applications web
- ♦ Apprendre les stratégies et les techniques de test des logiciels, les facteurs de qualité des logiciels et les différentes mesures utilisées

Module 7. Programmation Avancée

- ◆ Approfondir les connaissances en programmation, en particulier en ce qui concerne la programmation orientée objet et les différents types de relations entre les classes existantes
- ◆ Connaître les différents modèles de conception pour les problèmes concernant l'objet
- ◆ Apprendre la programmation événementielle et le développement d'interfaces utilisateurs avec Qt
- ◆ Acquérir les connaissances essentielles de la programmation concurrente, des processus et des fils d'exécution
- ◆ Apprendre à gérer l'utilisation des threads et de la synchronisation, ainsi que la résolution des problèmes courants de la programmation concurrente
- ◆ Comprendre l'importance de la documentation et des tests dans le développement de logiciels

Module 8. Réutilisation des Logiciels

- ◆ Obtenir une vue d'ensemble de la stratégie de réutilisation des logiciels
- ◆ Apprendre les différents modèles liés à la réutilisation des logiciels, à la fois en termes de conception, de création, de structure et de comportement
- ◆ Introduire le concept de *Framework*, et connaître les principaux types tels que ceux destinés à la conception d'interfaces graphiques, au développement d'applications web et à la gestion de la persistance des objets dans les bases de données
- ◆ Comprendre le fonctionnement du modèle MVC (Model-View-Controller) qui est actuellement très utilisé

Module 9. Intelligence Artificielle et Ingénierie des Connaissances

- ◆ Poser les bases de l'intelligence artificielle et de l'ingénierie de la connaissance, en faisant un bref retour sur l'histoire de l'intelligence artificielle jusqu'à aujourd'hui
- ◆ Comprendre les concepts essentiels de la recherche en intelligence artificielle, tant la recherche informée que non informée
- ◆ Comprendre comment l'intelligence artificielle fonctionne dans les jeux
- ◆ Apprendre les concepts fondamentaux des réseaux neuronaux et l'utilisation des algorithmes génétiques
- ◆ Acquérir les mécanismes appropriés pour représenter les connaissances, notamment dans la perspective du web sémantique
- ◆ Comprendre le fonctionnement des systèmes experts et des systèmes d'aide à la décision les systèmes d'aide à la décision

Module 10. Génie Logiciel Avancée

- ◆ Acquérir une connaissance approfondie des différentes méthodologies agiles utilisées dans le génie logiciel
- ◆ Apprendre à développer en utilisant Scrum, la programmation extrême et les techniques de développement logiciel basées sur la réutilisation
- ◆ Comprendre les différents modèles d'architectures de systèmes et de conception de logiciels, ainsi que l'architecture des applications en nuage
- ◆ Apprendre à réaliser des tests de logiciels, avec des méthodologies telles que *Test Driven Development*, *Acceptance Test Driven Development*, *Behavior Driven Development*, BDD et *Cucumber*
- ◆ Approfondir l'amélioration du processus de développement de logiciels et de la qualité des logiciels en utilisant les normes ISO/IEC
- ◆ Introduire le concept de DevOps et les principales pratiques qui en découlent

03

Compétences

Ce programme en Création des Interfaces et Applications en Réseau est conçu comme un outil de formation de haut niveau pour les professionnels. Le programme intensif contribuera de manière significative au développement et au travail des programmeurs et des portails web, en tirant parti des ressources Internet existantes et des logiciels libres, de manière pratique et utile.





“

Grâce aux compétences acquises dans le cadre de ce Mastère Spécialisé, vous pourrez vous lancer dans le développement d'applications et travailler en tant qu'ingénieur logiciel, avec la garantie d'une formation complète et actualisée"



Compétences générales

- ◆ Acquérir les compétences nécessaires à la pratique professionnelle de l'ingénierie informatique avec la connaissance de tous les facteurs nécessaires pour la mener à bien avec qualité et solvabilité

“

Une opportunité créée pour les professionnels qui recherchent un programme intensif et efficace, pour donner un élan significatif dans leur profession”





Compétences spécifiques

- ◆ Avoir une connaissance approfondie de toutes les facettes de l'interaction homme-machine et de la manière dont elles impliquent les développements informatiques
- ◆ Maîtriser l'utilisation des bases de données
- ◆ Développer les différents types d'applications en réseau
- ◆ Décrire et tirer parti des logiciels libres et des connaissances ouvertes sur le web
- ◆ Travailler en tant qu'ingénieur logiciel
- ◆ Contrôler l'utilisation de bases de données avancées
- ◆ Effectuer de la programmation avancée
- ◆ Savoir réutiliser les logiciels
- ◆ Créer des interfaces et des applications en réseau
- ◆ Maîtriser les différents systèmes de travail en génie logiciel avancé

04

Structure et contenu

Le contenu de ce Mastère Spécialisé a été élaboré par les différents experts de cette formation, avec un objectif clair: faire en sorte que nos étudiants acquièrent chacune des compétences nécessaires pour devenir de véritables experts dans ce domaine.

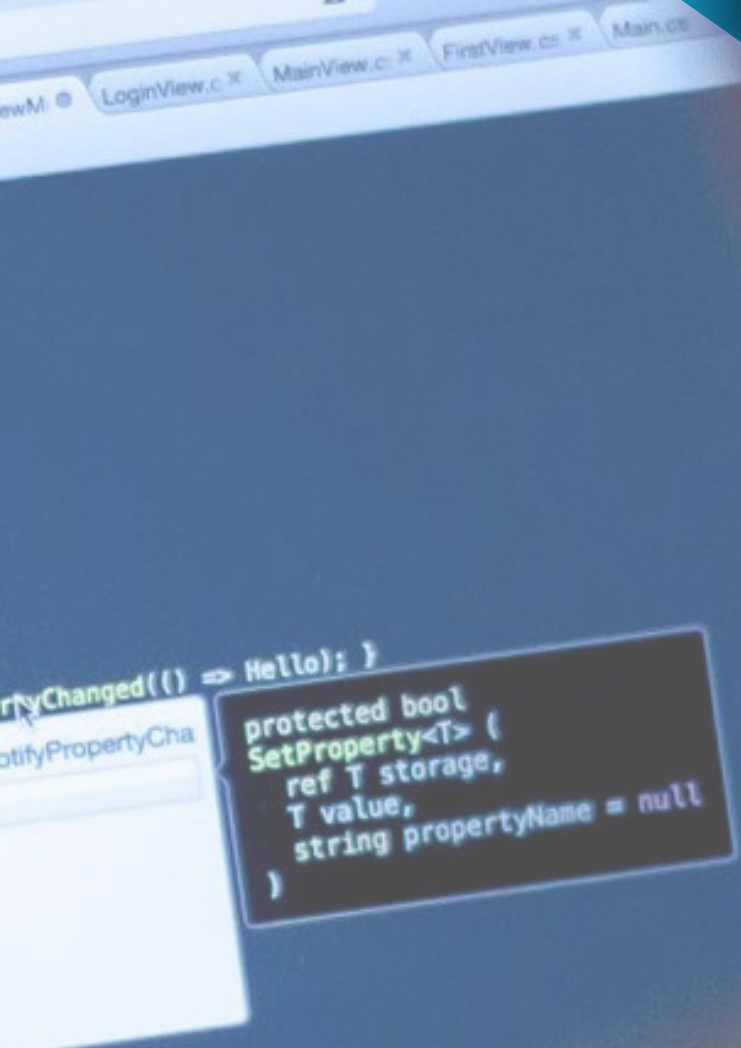
Un programme très complet et bien structuré qui vous mènera vers les plus hauts standards de qualité et de réussite.



```
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

public class MainViewModel : MvxViewModel
{
    public MainViewModel ()
    {
        private string _hello = "Hello MOFO";
        public string Hello
        {
            get { return _hello; }
            set { set _hello = value; RaisePrope
        }
    }
}

Cirrious.MvvmCross.ViewModels.MvxN
SetProperty
```



“

Un programme d'enseignement très complet, structuré en unités didactiques complètes et sur la base des dernières avancées du domaine, et orienté vers un apprentissage compatible avec votre vie personnelle et professionnelle"

Module 1. Interaction homme-ordinateur

- 1.1. Introduction à l'interaction homme-machine
 - 1.1.1. Qu'est-ce que l'interaction homme-machine?
 - 1.1.2. Relations entre l'interaction homme-machine et d'autres disciplines
 - 1.1.3. L'interface utilisateur
 - 1.1.4. Utilisabilité et accessibilité
 - 1.1.5. Expérience de l'utilisateur et conception centrée sur l'utilisateur
- 1.2. L'ordinateur et l'interaction: interface utilisateur et paradigmes d'interaction
 - 1.2.1. L'interaction
 - 1.2.2. Paradigmes et styles d'interaction
 - 1.2.3. Évolution des interfaces utilisateurs
 - 1.2.4. Interfaces utilisateur classiques: WIMP/GUI, commandes, voix, réalité virtuelle
 - 1.2.5. Interfaces utilisateurs innovantes: portables, collaboratives, BCI
- 1.3. Le facteur humain: aspects psychologiques et cognitifs
 - 1.3.1. L'importance du facteur humain dans l'interaction
 - 1.3.2. Le traitement de l'information par l'homme
 - 1.3.3. L'entrée et la sortie de l'information: visuelle, auditive et tactile
 - 1.3.4. Perception et attention
 - 1.3.5. Connaissances et modèles mentaux: représentation, organisation et acquisition
- 1.4. Le facteur humain: limites sensorielles et physiques
 - 1.4.1. Diversité fonctionnelle, handicap et déficience
 - 1.4.2. Diversité visuelle
 - 1.4.3. Diversité auditive
 - 1.4.4. Diversité cognitive
 - 1.4.5. Diversité motrice
 - 1.4.6. Le cas des immigrants numériques
- 1.5. Le processus de conception (I): analyse des besoins pour la conception de l'interface utilisateur
 - 1.5.1. Conception centrée sur l'utilisateur
 - 1.5.2. Qu'est-ce que l'analyse des besoins?
 - 1.5.3. Collecte d'informations
 - 1.5.4. Analyse et interprétation des informations
 - 1.5.5. Analyse de la facilité d'utilisation et de l'accessibilité



- 1.6. Le processus de conception (II): prototypage et analyse des tâches
 - 1.6.1. Design conceptuel
 - 1.6.2. Prototype
 - 1.6.3. Analyse hiérarchique des tâches
 - 1.7. Le processus de conception (III): évaluation
 - 1.7.1. L'évaluation dans le processus de conception: objectifs et méthodes
 - 1.7.2. Méthodes d'évaluation sans utilisateurs
 - 1.7.3. Méthodes d'évaluation avec les utilisateurs
 - 1.7.4. Standards et normes d'évaluation
 - 1.8. Accessibilité: définition et lignes directrices
 - 1.8.1. Accessibilité et conception universelle
 - 1.8.2. L'initiative WAI et les lignes directrices WCAG
 - 1.8.3. Directives WCAG 2.0 et 2.1
 - 1.9. Accessibilité: évaluation et diversité fonctionnelle
 - 1.9.1. Outils d'évaluation de l'accessibilité du web
 - 1.9.2. Accessibilité et diversité fonctionnelle
 - 1.10. L'ordinateur et l'interaction: périphériques et dispositifs
 - 1.10.1. Dispositifs et périphériques traditionnels
 - 1.10.2. Dispositifs et périphériques alternatifs
 - 1.10.3. Téléphones mobiles et tablettes
 - 1.10.4. Diversité fonctionnelle, interaction et périphériques
-
- Module 2. Bases de données**
- 2.1. Applications et finalités des systèmes de base de données
 - 2.1.1. Applications des différents systèmes de base de données
 - 2.1.2. Objectif dans les différents systèmes de base de données
 - 2.1.3. Vision des données
 - 2.2. Base de données et architecture
 - 2.2.1. Base de données relationnelle
 - 2.2.2. Conception de la base de données
 - 2.2.3. Bases de données à base d'objets et semi-structurées
 - 2.2.4. Stockage des données et consultations
 - 2.2.5. Gestion des transactions
 - 2.2.6. Exploration et analyse des données
 - 2.2.7. Architecture des bases de données
 - 2.3. Le modèle relationnel: structure, opérations et algèbre relationnelle étendue
 - 2.3.1. La structure des BD relationnelles
 - 2.3.2. Opérations fondamentales de l'algèbre relationnelle
 - 2.3.3. Autres opérations de l'algèbre relationnelle
 - 2.3.4. Opérations étendues de l'algèbre relationnelle
 - 2.3.5. Valeurs nulles
 - 2.3.6. Modification de la base de données
 - 2.4. SQL (I)
 - 2.4.1. Qu'est-ce que SQL?
 - 2.4.2. La définition des données
 - 2.4.3. La structure de base des requêtes SQL
 - 2.4.4. Opérations sur les ensembles
 - 2.4.5. Fonctions d'agrégation
 - 2.4.6. Valeurs nulles
 - 2.5. SQL (II)
 - 2.5.1. Sous-requêtes imbriquées
 - 2.5.2. Consultations complexes
 - 2.5.3. Vues
 - 2.5.4. Curseurs
 - 2.5.5. Consultations complexes
 - 2.5.6. Déclencheurs
 - 2.6. Conception de la base de données et modèle E-R
 - 2.6.1. Aperçu du processus de conception
 - 2.6.2. Le modèle entité-relation
 - 2.6.3. Restrictions
 - 2.7. Diagrammes entité-relation
 - 2.7.1. Diagrammes entité-relation
 - 2.7.2. Aspects de la conception des relations entre entités
 - 2.7.3. Ensembles d'entités faibles
 - 2.8. Le modèle entité-relation étendu
 - 2.8.1. Caractéristiques du modèle E-R étendu
 - 2.8.2. Conception d'une base de données
 - 2.8.3. Réduction aux schémas relationnels

- 2.9. Conception d'une base de données relationnelle
 - 2.9.1. Caractéristiques d'une bonne conception relationnelle
 - 2.9.2. Domaines atomiques et première forme normale (1FN)
 - 2.9.3. Décomposition à l'aide de dépendances fonctionnelles
 - 2.9.4. Théorie des dépendances fonctionnelles
 - 2.9.5. Algorithmes de décomposition
 - 2.9.6. Décomposition à l'aide de dépendances multivaluées
 - 2.9.7. Autres formes normales
 - 2.9.8. Processus de conception des bases de données
- 2.10. Bases de données NoSQL
 - 2.10.1. Qu'est-ce qu'une base de données NoSQL?
 - 2.10.2. Analyse des différentes options NoSQL et de leurs caractéristiques
 - 2.10.3. Mongo DB

Module 3. Développement d'applications en réseau

- 3.1. Langages de balisage HTML5
 - 3.1.1. Concepts de base de HTML
 - 3.1.2. Nouveaux éléments HTML 5
 - 3.1.3. Formulaire: nouveaux contrôles
- 3.2. Introduction aux feuilles de style CSS
 - 3.2.1. Premiers pas avec CSS
 - 3.2.2. Introduction à CSS3
- 3.3. Langage de script du navigateur: JavaScript
 - 3.3.1. Concepts de base de JavaScript
 - 3.3.2. DOM
 - 3.3.3. Événements
 - 3.3.4. JQuery
 - 3.3.5. Ajax
- 3.4. Concept de programmation des composants
 - 3.4.1. Contexte
 - 3.4.2. Composants et interfaces
 - 3.4.3. États d'un composant

- 3.5. Architecture des composants
 - 3.5.1. Architectures actuelles
 - 3.5.2. Intégration et déploiement des composants
- 3.6. *Framework Front-End: Bootstrap*
 - 3.6.1. Conception de la grille
 - 3.6.2. Formulaire
 - 3.6.3. Composants
- 3.7. Modèle de vue du contrôleur
 - 3.7.1. Méthodes de développement web
 - 3.7.2. Modèles de conception: MVC
- 3.8. Technologies *Grid* de l'Information
 - 3.8.1. Augmentation des ressources informatiques
 - 3.8.2. Concept de la technologie *Grid*
- 3.9. Architectures orientées vers les services
 - 3.9.1. SOA et services web
 - 3.9.2. Topologie des services web
 - 3.9.3. Plateformes de services web
- 3.10. Protocole HTTP
 - 3.10.1. Messages
 - 3.10.2. Sessions persistantes
 - 3.10.3. Système cryptographique
 - 3.10.4. Fonctionnement du protocole HTTPS

Module 4. Logiciels libres et connaissances ouvertes

- 4.1. Introduction aux logiciels libres
 - 4.1.1. Histoire du logiciel libre
 - 4.1.2. La "liberté" dans les logiciels
 - 4.1.3. Licences d'utilisation des outils logiciels
 - 4.1.4. Propriété intellectuelle des logiciels
 - 4.1.5. Quelle est la motivation pour utiliser un logiciel libre?
 - 4.1.6. Les mythes du logiciel libre
 - 4.1.7. Top500

- 4.2. Connaissances ouvertes et licences CC
 - 4.2.1. Concepts de base
 - 4.2.2. Licences *Creative Commons*
 - 4.2.3. Autres licences de contenu
 - 4.2.4. Wikipédia et autres projets de connaissance ouverte
- 4.3. Principaux outils logiciels libres
 - 4.3.1. Systèmes d'exploitation
 - 4.3.2. Applications bureautiques
 - 4.3.3. Applications de gestion d'entreprise
 - 4.3.4. Gestionnaires de contenu Web
 - 4.3.5. Outils de création de contenu multimédia
 - 4.3.6. Autres applications
- 4.4. L'entreprise: le logiciel libre et ses coûts
 - 4.4.1. Logiciel libre: oui ou non?
 - 4.4.2. Vérités et mensonges sur les logiciels libres
 - 4.4.3. Logiciels d'entreprise basés sur des logiciels libres
 - 4.4.4. Coûts des logiciels
 - 4.4.5. Modèles de software libre
- 4.5. Le système d'exploitation GNU/Linux
 - 4.5.1. Architecture
 - 4.5.2. Structure de base des répertoires
 - 4.5.3. Caractéristiques et structure du système de fichiers
 - 4.5.4. Représentation interne des dossiers
- 4.6. Le système d'exploitation mobile Android
 - 4.6.1. Histoire
 - 4.6.2. Architecture
 - 4.6.3. Forks de Android
 - 4.6.4. Introduction au développement Android
 - 4.6.5. *Frameworks* pour le développement d'applications mobiles
- 4.7. Créer des sites web avec WordPress
 - 4.7.1. Caractéristiques et structure de WordPress
 - 4.7.2. Création de sites wordpress.com
 - 4.7.3. Installation et configuration de WordPress sur votre propre serveur
 - 4.7.4. Installation de plugins et extension de WordPress
 - 4.7.5. Création de plugins WordPress
 - 4.7.6. Création de thèmes WordPress
- 4.8. Création de thèmes WordPress
 - 4.8.1. Environnements en nuage
 - 4.8.2. Outils de suivi
 - 4.8.3. Systèmes d'exploitation
 - 4.8.4. Big Data et données ouvertes 2.0
 - 4.8.5. L'informatique quantique
- 4.9. Contrôle des versions
 - 4.9.1. Concepts de base
 - 4.9.2. Git
 - 4.9.3. Services Git dans le cloud et autohébergé
 - 4.9.4. Autres systèmes de contrôle de version
- 4.10. Distributions GNU/Linux personnalisées
 - 4.10.1. Principales distributions
 - 4.10.2. Distributions dérivées de Debian
 - 4.10.3. Création de paquets deb
 - 4.10.4. Modifier la distribution
 - 4.10.5. Générer des images ISO

Module 5. Bases de Données Avancées

- 5.1. Introduction aux différents systèmes de base de données
 - 5.1.1. Rappel historique
 - 5.1.2. Bases de données hiérarchique
 - 5.1.3. Bases de données web
 - 5.1.4. Base de données relationnelles
 - 5.1.5. Base de données non relationnelles

- 5.2. XML et bases de données pour le web
 - 5.2.1. Validation des documents XML
 - 5.2.2. Transformations des documents XML
 - 5.2.3. Stockage des données XML
 - 5.2.4. Base de données relationnelles XML
 - 5.2.5. SQL/XML
 - 5.2.6. Base de données natives XML
- 5.3. Base de données parallèles
 - 5.3.1. Systèmes parallèles
 - 5.3.2. Architectures de bases de données parallèles
 - 5.3.3. Parallélisme des requêtes
 - 5.3.4. Parallélisme des requêtes
 - 5.3.5. Conception de systèmes parallèles
 - 5.3.6. Traitement parallèle dans SQL
- 5.4. Base de données distribués
 - 5.4.1. Systèmes distribués
 - 5.4.2. Stockage distribué
 - 5.4.3. Disponibilité
 - 5.4.4. Traitement distribué des requêtes
 - 5.4.5. Fournisseurs de bases de données distribués
- 5.5. Indexation et association
 - 5.5.1. Index ordonnés
 - 5.5.2. Index denses et épars
 - 5.5.3. Index multiniveaux
 - 5.5.4. Actualisation des index
 - 5.5.5. Association statique
 - 5.5.6. Comment utiliser les index dans les bases de données
- 5.6. Introduction au traitement transactionnel
 - 5.6.1. Les états d'une transaction
 - 5.6.2. Implémentation de l'atomicité et de la durabilité
 - 5.6.3. Séquentialité
 - 5.6.4. Récupérabilité
 - 5.6.5. Mise en œuvre de l'isolement



- 5.7. Systèmes de récupération
 - 5.7.1. Classification des défaillances
 - 5.7.2. Structures de stockage
 - 5.7.3. Récupération et atomicité
 - 5.7.4. Récupération sur la base d'un enregistrement historique
 - 5.7.5. Transactions simultanées et récupération
 - 5.7.6. Haute disponibilité des bases de données
- 5.8. Exécution et traitement des consultations
 - 5.8.1. Coût d'une requête
 - 5.8.2. Opération de sélection
 - 5.8.3. Le tri
 - 5.8.4. Introduction à l'optimisation des requêtes
 - 5.8.5. Contrôle des performances
- 5.9. Base de données non relationnelles
 - 5.9.1. Bases de données axées sur les documents
 - 5.9.2. Bases de données axées sur les graphes
 - 5.9.3. Bases de données clé-valeur
- 5.10. Data Warehouse, OLAP et Data Mining
 - 5.10.1. Composants de stockage de données
 - 5.10.2. Architecture du Data Warehouse
 - 5.10.3. OLAP
 - 5.10.4. Fonctionnalités d'exploration des données
 - 5.10.5. Autres types d'exploration

Module 6. Génie Logiciel

- 6.1. Cadre du génie logiciel
 - 6.1.1. Caractéristiques des logiciels
 - 6.1.2. Principaux processus du génie logiciel
 - 6.1.3. Modèles de processus de développement de logiciels
 - 6.1.4. Cadre de référence standard pour le développement de logiciels: la norme ISO/IEC 12207
- 6.2. Processus unifiés de développement logiciels
 - 6.2.1. Processus unifié
 - 6.2.2. Dimensions du processus unifié
 - 6.2.3. Processus de développement axé sur les cas d'utilisation
 - 6.2.4. Flux de travail fondamentaux du processus unifié

- 6.3. Planification dans le contexte du développement logiciels agiles
 - 6.3.1. Caractéristiques du développement de logiciels agiles
 - 6.3.2. Différents horizons de planification dans le cadre du développement agile
 - 6.3.3. Cadre de développement agile Scrum et horizons de planification
 - 6.3.4. Histoires d'utilisateurs comme unité de planification et d'estimation
 - 6.3.5. Techniques courantes pour obtenir une estimation
 - 6.3.6. Échelles d'interprétation des estimations
 - 6.3.7. *Planning poker*
 - 6.3.8. Types d'ordonnancement courants: planification des livraisons et planification des itérations
- 6.4. Styles de conception de logiciels distribués et architectures logicielles orientées services
 - 6.4.1. Modèles de communication dans les systèmes logiciels distribués
 - 6.4.2. Couche intermédiaire ou middleware
 - 6.4.3. Modèles d'architecture pour les systèmes distribués
 - 6.4.4. Processus général de conception de services logiciels
 - 6.4.5. Aspects de la conception de services logiciels
 - 6.4.6. Composition des services
 - 6.4.7. Architecture des services web
 - 6.4.8. Composants de l'infrastructure et SOA
- 6.5. Introduction au développement de logiciels basés sur des modèles
 - 6.5.1. Le concept de modèle
 - 6.5.2. Développement de logiciels piloté par les modèles
 - 6.5.3. Cadre de développement piloté par le modèle MDA
 - 6.5.4. Éléments d'un modèle de transformation
- 6.6. Conception de l'Interface Graphique de l'Utilisateur
 - 6.6.1. Principes de conception de l'interface utilisateur
 - 6.6.2. Modèles de conception architecturale pour les systèmes interactifs: Modèle Vue Contrôleur (MVC)
 - 6.6.3. Expérience de l'utilisateur (UX *User Experience*)
 - 6.6.4. Conception centrée sur l'utilisateur
 - 6.6.5. Analyse de l'interface utilisateur graphique et processus de conception
 - 6.6.6. Facilité d'utilisation des interfaces utilisateur
 - 6.6.7. Accessibilité des interfaces utilisateurs
- 6.7. Conception d'applications web
 - 6.7.1. Caractéristiques des applications web
 - 6.7.2. Interface utilisateur de l'application web
 - 6.7.3. Conception de la navigation
 - 6.7.4. Protocole d'interaction de base pour les applications web
 - 6.7.5. Styles d'architecture pour les applications web
- 6.8. Stratégies et techniques de test des logiciels et facteurs de qualité des logiciels
 - 6.8.1. Stratégies de test
 - 6.8.2. Conception des cas de test
 - 6.8.3. Rapport qualité-prix
 - 6.8.4. Modèles de qualité
 - 6.8.5. Famille de normes ISO/IEC 25000 (SQuaRE)
 - 6.8.6. Modèle de qualité des produits (ISO 2501n)
 - 6.8.7. Modèles de qualité des données (ISO 2501n)
 - 6.8.8. Gestion de la qualité des logiciels
- 6.9. Introduction aux métriques du génie logiciel
 - 6.9.1. Concepts de base: mesures, métriques et indicateurs
 - 6.9.2. Types de métriques de génie logiciel
 - 6.9.3. Le processus de mesure
 - 6.9.4. ISO 25024 Mesures externes et mesures de la qualité utilisées
 - 6.9.5. Mesures orientées à l'objet
- 6.10. Maintenance et réingénierie des logiciels
 - 6.10.1. Processus de maintenance
 - 6.10.2. Cadre du processus de maintenance standard ISO/EIEC 14764
 - 6.10.3. Modèle de processus de réingénierie des logiciels
 - 6.10.4. Ingénierie inverse

Module 7. Programmation Avancée

- 7.1. Introduction à la programmation des objets
 - 7.1.1. Introduction à la programmation des objets
 - 7.1.2. Conception des classes
 - 7.1.3. Introduction à UML pour la modélisation des problèmes
- 7.2. Relations entre les classes
 - 7.2.1. Abstraction et héritage
 - 7.2.2. Concepts avancés d'héritage
 - 7.2.3. Polymorphisme
 - 7.2.4. Composition et agrégation
- 7.3. Introduction aux modèles de conception pour les problèmes orientés objet
 - 7.3.1. Que sont les patrons de conception?
 - 7.3.2. Modèle *Factory*
 - 7.3.3. Modèle Singleton
 - 7.3.4. Modèle *Observer*
 - 7.3.5. Modèle *Composite*
- 7.4. Exceptions
 - 7.4.1. Qu'est-ce qu'une exception?
 - 7.4.2. Capture et gestion des exceptions
 - 7.4.3. Lancer des exceptions
 - 7.4.4. Création d'une exception
- 7.5. Interfaces utilisateur
 - 7.5.1. Introduction à Qt
 - 7.5.2. Positionnement
 - 7.5.3. Qu'est-ce qu'un événement?
 - 7.5.4. Événements: définition et capture
 - 7.5.5. Développement d'interfaces utilisateurs
- 7.6. Introduction à la programmation simultanée
 - 7.6.1. Introduction à la programmation simultanée
 - 7.6.2. Le concept de processus et de thread
 - 7.6.3. Interaction entre processus ou threads
 - 7.6.4. Les threads en C++
 - 7.6.5. Avantages et inconvénients de la programmation concurrente
- 7.7. Gestion des threads et synchronisation
 - 7.7.1. Cycle de vie des threads
 - 7.7.2. La classe Thread
 - 7.7.3. L'ordonnement des threads
 - 7.7.4. Les groupes de threads
 - 7.7.5. Fils de type démon
 - 7.7.6. Synchronisation
 - 7.7.7. Mécanismes de verrouillage
 - 7.7.8. Mécanismes de communication
 - 7.7.9. Moniteurs
- 7.8. Problèmes courants dans la programmation concurrente
 - 7.8.1. Le problème du producteur-consommateur
 - 7.8.2. Le problème des lecteurs et des écrivains
 - 7.8.3. Le problème du dîner des philosophes
- 7.9. Documentation et tests de logiciels
 - 7.9.1. Pourquoi est-il important de documenter les logiciels?
 - 7.9.2. Documentation de la conception
 - 7.9.3. Utilisation d'outils pour la documentation
- 7.10. Tests de logiciels
 - 7.10.1. Introduction aux tests de logiciels
 - 7.10.2. Types de tests
 - 7.10.3. Tests unitaires
 - 7.10.4. Tests d'intégration
 - 7.10.5. Test de validation
 - 7.10.6. Test du système

Module 8. Réutilisation des Logiciels

- 8.1. Panorama de la réutilisation des logiciels
 - 8.1.1. Qu'est-ce que la réutilisation des logiciels?
 - 8.1.2. Avantages et inconvénients de la réutilisation des logiciels
 - 8.1.3. Principales techniques de réutilisation des logiciels
- 8.2. Introduction aux modèles de conception
 - 8.2.1. Qu'est-ce qu'un modèle de conception?
 - 8.2.2. Catalogue des principaux patrons de conception
 - 8.2.3. Comment utiliser les modèles pour résoudre des problèmes de conception
 - 8.2.4. Comment sélectionner le meilleur modèle de conception
- 8.3. Les modèles de création
 - 8.3.1. Les modèles de création
 - 8.3.2. Modèle *Abstract Factory*
 - 8.3.3. Exemple d'implémentation du modèle *Abstract Factory*
 - 8.3.4. Modèle Builder
 - 8.3.5. Exemple d'implémentation de Builder
 - 8.3.6. Modèle *Abstract Factory* vs Builder
- 8.4. Les modèles de création(II)
 - 8.4.1. Modèle *Factory Method*
 - 8.4.2. *Factory Method* vs. *Abstract Factory*
 - 8.4.3. Modèle Singleton
- 8.5. Modèles structurels
 - 8.5.1. Modèles structurels
 - 8.5.2. Modèle *Adapter*
 - 8.5.3. Modèle *Bridge*
- 8.6. Les modèles structurelles (II)
 - 8.6.1. Modèle *Composite*
 - 8.6.2. Modèle de décorateur
- 8.7. Les modèles structurelles (III)
 - 8.7.1. Modèle Facade
 - 8.7.2. Modèle Proxy
- 8.8. Modèles de comportement
 - 8.8.1. Concept de modèle de comportement
 - 8.8.2. Modèle de comportement: chaîne de responsabilité
 - 8.8.3. Ordre des schémas de comportement

- 8.9. Modèles de comportement (II)
 - 8.9.1. Modèle d'interprétation ou *Interpreter*
 - 8.9.2. Modèle d'itérateur
 - 8.9.3. Modèle d'observateur
 - 8.9.4. Modèle de stratégie
- 8.10. *Frameworks*
 - 8.10.1. Modèle de *Frameworks*
 - 8.10.2. Développement à l'aide *Frameworks*
 - 8.10.3. Modèle *Model View Controller*
 - 8.10.4. *Framework* pour la conception d'une interface graphique de l'utilisateur
 - 8.10.5. *Frameworks* pour le développement d'applications web
 - 8.10.6. *Frameworks* pour la gestion de la persistance des objets dans les bases de données

Module 9. Intelligence Artificielle et Ingénierie des Connaissances

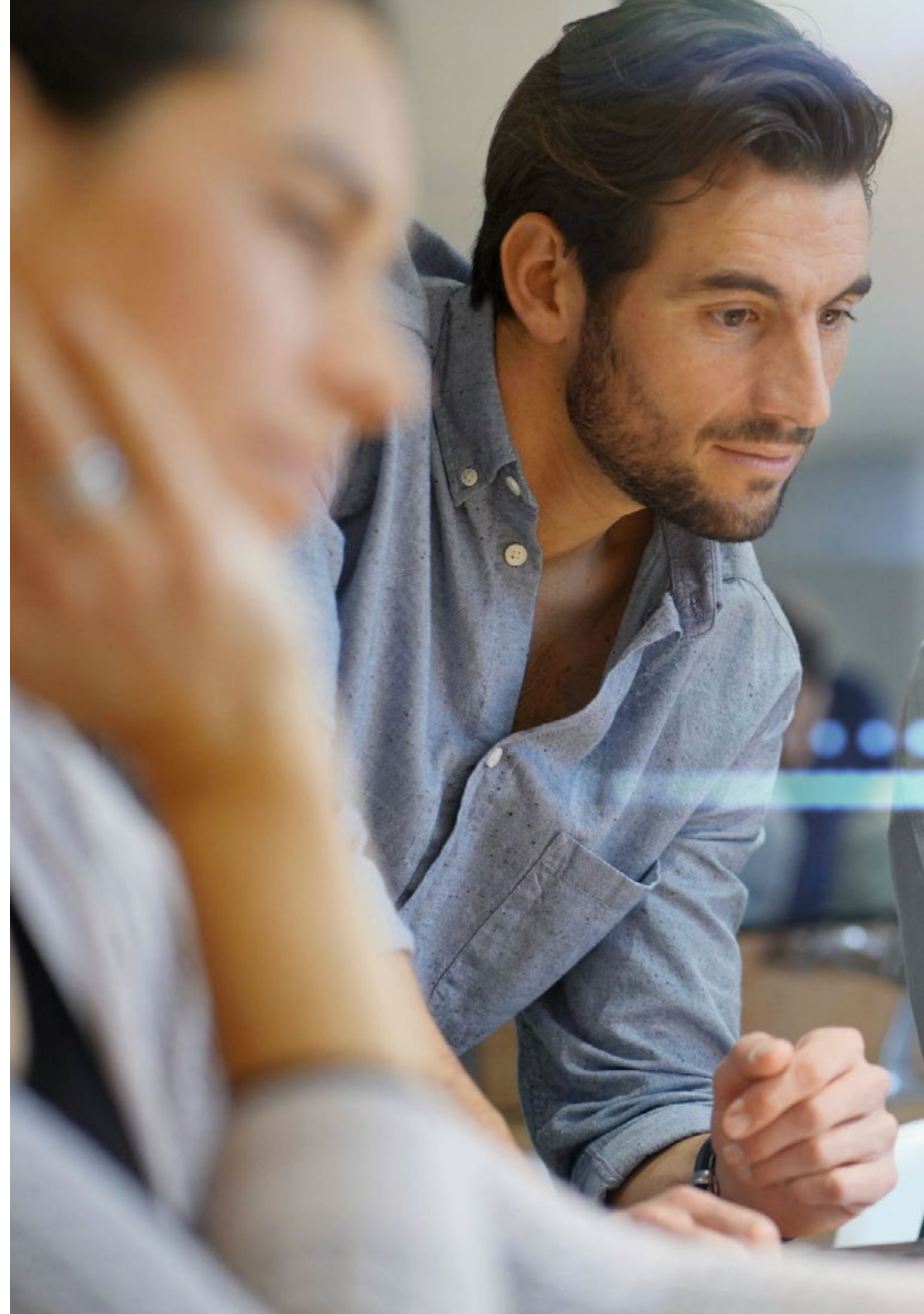
- 9.1. Introduction à l'Intelligence Artificielle et à l'Ingénierie des Connaissances
 - 9.1.1. Brève histoire de l'Intelligence Artificielle
 - 9.1.2. L'Intelligence Artificielle aujourd'hui
 - 9.1.3. Ingénierie de la connaissance
- 9.2. Recherche
 - 9.2.1. Concepts de recherche courants
 - 9.2.2. Recherche non informée
 - 9.2.3. Recherche informée
- 9.3. Satisfaisabilité booléenne, satisfaisabilité des contraintes et planification automatique
 - 9.3.1. Satisfaisabilité booléenne
 - 9.3.2. Problèmes de satisfaction des contraintes
 - 9.3.3. Planification automatique et PDDL
 - 9.3.4. La planification comme recherche heuristique
 - 9.3.5. Planification avec le SAT
- 9.4. L'Intelligence Artificielle dans les jeux
 - 9.4.1. Théorie des jeux
 - 9.4.2. Minimax et Alpha-Beta
 - 9.4.3. Simulation: Monte Carlo

- 9.5. Apprentissage supervisé et non supervisé
 - 9.5.1. Introduction à l'apprentissage automatique
 - 9.5.2. Classification
 - 9.5.3. Régression
 - 9.5.4. Validation des résultats
 - 9.5.5. Regroupement (*Clustering*)
- 9.6. Réseaux neuronaux
 - 9.6.1. Fondements biologiques
 - 9.6.2. Modèle computationnel
 - 9.6.3. Réseaux neuronaux supervisés et non supervisés
 - 9.6.4. Perceptron simple
 - 9.6.5. Perceptron multicouche
- 9.7. Algorithmes génétiques
 - 9.7.1. Histoire
 - 9.7.2. Base biologique
 - 9.7.3. Codification des problèmes
 - 9.7.4. Génération de la population initiale
 - 9.7.5. Algorithme principal et opérateurs génétiques
 - 9.7.6. Évaluation des individus: Fitness
- 9.8. Thésaurus, vocabulaires, taxonomies
 - 9.8.1. Vocabulaires
 - 9.8.2. Taxonomies
 - 9.8.3. Thésaurus
 - 9.8.4. Ontologies
- 9.9. Représentation des connaissances: web sémantique
 - 9.9.1. Web sémantique
 - 9.9.2. Spécifications: RDF, RDFS et OWL
 - 9.9.3. Inférence/raisonnement
 - 9.9.4. *Linked Data*
- 9.10. Systèmes experts et DSS
 - 9.10.1. Systèmes experts
 - 9.10.2. Systèmes d'aide à la décision

Module 10. Génie Logiciel Avancée

- 10.1. Introduction aux méthodologies agiles
 - 10.1.1. Modèles de processus et méthodologies
 - 10.1.2. Agilité et processus agiles
 - 10.1.3. Manifeste agile
 - 10.1.4. Quelques méthodologies agiles
 - 10.1.5. Agile vs. Traditionnel
- 10.2. Scrum
 - 10.2.1. Origine et philosophie de Scrum
 - 10.2.2. Valeurs de Scrum
 - 10.2.3. Déroulement du processus Scrum
 - 10.2.4. Rôles de Scrum
 - 10.2.5. Les artéfacts Scrum
 - 10.2.6. Les événements Scrum
 - 10.2.7. Les histoires d'utilisateurs
 - 10.2.8. Extensions de Scrum
 - 10.2.9. Estimations agiles
 - 10.2.10. Scrum Scaling
- 10.3. Programmation extrême
 - 10.3.1. Raison d'être et aperçu de XP
 - 10.3.2. Le cycle de vie XP
 - 10.3.3. Les cinq valeurs fondamentales
 - 10.3.4. Les douze pratiques de base de XP
 - 10.3.5. Les rôles des participants
 - 10.3.6. Le XP industriel
 - 10.3.7. Évaluation critique de XP
- 10.4. Développement de logiciels basés sur la réutilisation
 - 10.4.1. Réutilisation des logiciels
 - 10.4.2. Niveaux de réutilisation du code
 - 10.4.3. Techniques spécifiques de réutilisation
 - 10.4.4. Développement à base de composants
 - 10.4.5. Avantages et problèmes de la réutilisation
 - 10.4.6. Planification de la réutilisation

- 10.5. Architecture des systèmes et modèles de conception des logiciels
 - 10.5.1. Conception architecturale
 - 10.5.2. Modèles d'architecture généraux
 - 10.5.3. Architectures tolérantes aux défaillances
 - 10.5.4. Architectures de systèmes distribués
 - 10.5.5. Modèles de conception
 - 10.5.6. Modèles Gamma
 - 10.5.7. Modèles de conception d'interaction
- 10.6. Architecture des applications en nuage
 - 10.6.1. Principes fondamentaux du *Cloud Computing*
 - 10.6.2. Qualité des applications en nuage
 - 10.6.3. Styles d'architecture
 - 10.6.4. Modèles de conception
- 10.7. Tests de logiciels: TDD, ATDD y BDD
 - 10.7.1. Vérification et validation des logiciels
 - 10.7.2. Tests de logiciels
 - 10.7.3. *Tests Driven Development* (TDD)
 - 10.7.4. *Acceptance Tests Driven Development* (ATDD)
 - 10.7.5. *Behavior Driven Development* (BDD)
 - 10.7.6. BDD et *Cucumber*
- 10.8. Amélioration du processus logiciel
 - 10.8.1. Amélioration du processus logiciel
 - 10.8.2. Le processus d'amélioration du processus
 - 10.8.3. Modèles de maturité
 - 10.8.4. Le modèle CMMI
 - 10.8.5. CMMI V2.0
 - 10.8.6. CMMI et Agile





- 10.9. La qualité du produit logiciel: SQuaRE
 - 10.9.1. Qualité des logiciels
 - 10.9.2. Modèle de qualité du produit logiciel
 - 10.9.3. Famille ISO/ IEC 25000
 - 10.9.4. ISO/IEC 25010: modèle de qualité et caractéristiques de qualité
 - 10.9.5. ISO/IEC 25012: qualité des données
 - 10.9.6. ISO/IEC 25020: mesure de la qualité des logiciels
 - 10.9.7. ISO/IEC 25022, 25023 et 25024: mesures de la qualité des logiciels et des données
 - 10.9.8. ISO/CEI 25040: évaluation des logiciels
 - 10.9.9. Processus de certification
- 10.10. Introduction à DevOps
 - 10.10.1. Concept de DevOps
 - 10.10.2. Pratiques essentielles

“ Une expérience unique, clé et décisive pour stimuler votre développement professionnel”

05 Méthodologie

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

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



“

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

Étude de Cas pour mettre en contexte tout le contenu

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

“

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



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



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

Une méthode d'apprentissage innovante et différente

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

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

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

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

Relearning Methodology

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

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

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

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

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





Dans notre programme, l'apprentissage n'est pas un processus linéaire, mais se déroule en spirale (apprendre, désapprendre, oublier et réapprendre). Par conséquent, chacun de ces éléments est combiné de manière concentrique. Cette méthodologie a permis de former plus de 650.000 diplômés universitaires avec un succès sans précédent dans des domaines aussi divers que la biochimie, la génétique, la chirurgie, le droit international, les compétences en gestion, les sciences du sport, la philosophie, le droit, l'ingénierie, le journalisme, l'histoire, les marchés financiers et les instruments. Tout cela dans un environnement très exigeant, avec un corps étudiant universitaire au profil socio-économique élevé et dont l'âge moyen est de 43,5 ans.

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

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

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

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



Support d'étude

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

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



Cours magistraux

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

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



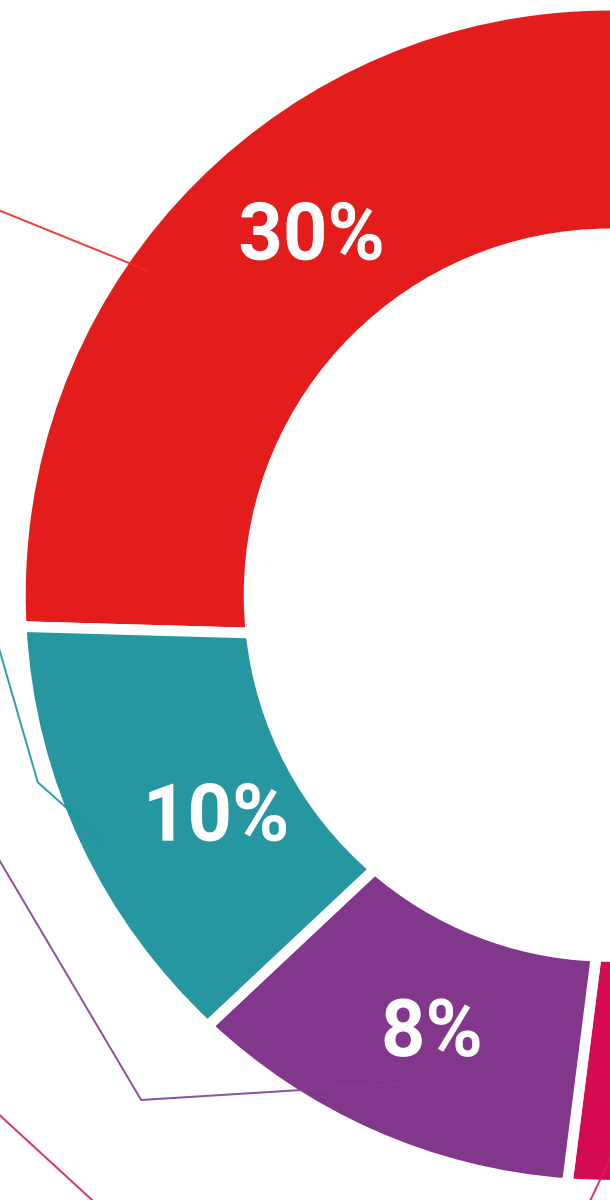
Pratiques en compétences et aptitudes

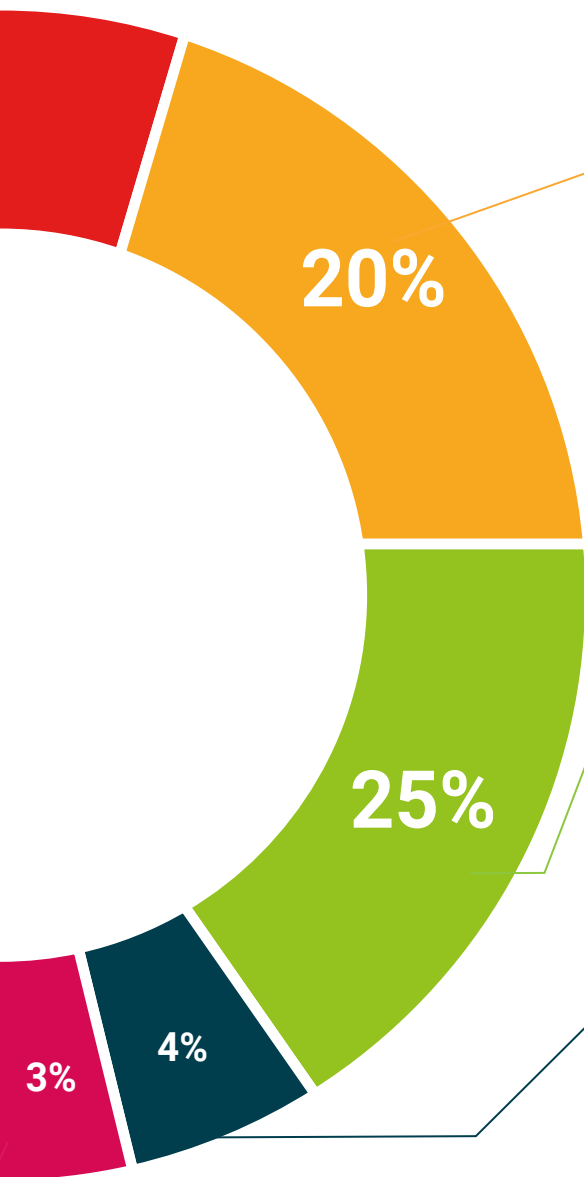
Les étudiants réaliseront des activités visant à développer des compétences et des aptitudes spécifiques dans chaque domaine. Des activités pratiques et dynamiques pour acquérir et développer les compétences et aptitudes qu'un spécialiste doit développer dans le cadre de la mondialisation dans laquelle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



Lectures complémentaires

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





Case studies

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



Résumés interactifs

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



Testing & Retesting

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



06 Diplôme

Le Mastère Spécialisé en Création des Interfaces et Applications en Réseau vous garantit, en plus de la formation la plus rigoureuse et la plus actuelle, l'accès à un diplôme universitaire de Mastère Spécialisé délivré par TECH Université Technologique.



“

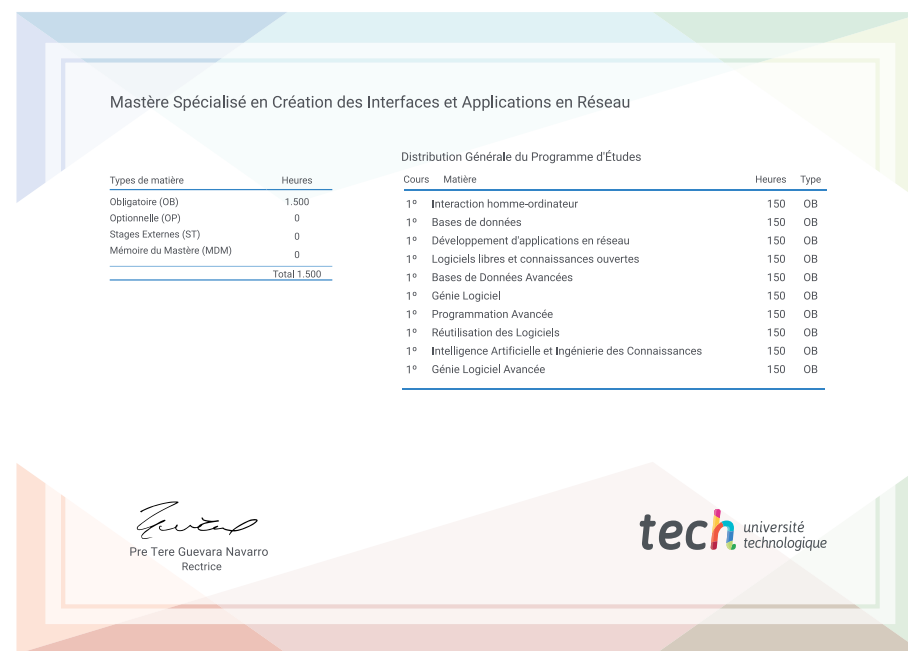
Terminez ce programme avec succès et recevez votre diplôme universitaire sans avoir à vous soucier des contraintes de déplacements ou des formalités administratives”

Ce **Mastère Spécialisé en Création des Interfaces et Applications en Réseau** contient le programme le plus complet et le plus à jour du marché.

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

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

Diplôme: **Mastère Spécialisé en Création des Interfaces et Applications en Réseau**
N° d'heures officielles: **1.500 h.**



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

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

tech université
technologique

Mastère Spécialisé Création des Interfaces et Applications en Réseau

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

Mastère Spécialisé

Création des Interfaces et Applications en Réseau