

Mastère Spécialisé

Programmation Front End Full Stack Developer





Mastère Spécialisé Programmation Front End Full Stack Developer

- » 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.techtute.com/fr/informatique/master/master-programmation-front-end-full-stack-developer

Sommaire

01

Présentation

page 4

02

Objectifs

page 8

03

Compétences

page 14

04

Direction de la formation

page 18

05

Structure et contenu

page 22

06

Méthodologie

page 34

07

Diplôme

page 42

01

Présentation

Les développeurs *Front End* sont une partie essentielle de tout développement web. La nature de l'union entre la partie conception graphique et la programmation *Back End* proprement dite leur impose de posséder des connaissances très spécifiques et développées, qui doivent à leur tour être mises à jour pour pouvoir faire face aux défis de plus en plus exigeants du présent. S'agissant d'une spécialité qui offre de multiples opportunités au niveau professionnel, TECH a réuni la meilleure équipe pédagogique possible pour développer un programme universitaire étendu, complet et méthodologique. En se concentrant sur les outils et les méthodologies les plus avancés dans le développement de solutions web, ce diplôme offre l'élan nécessaire pour franchir le pas vers la programmation *Front End Full Stack*.



“

Devenez un élément essentiel de tout développement web, en dirigeant et coordonnant des équipes informatiques grâce à la méthodologie de travail et de leadership perfectionnée de ce Mastère Spécialisé”

Le développement *Full Stack* est une option particulièrement intéressante pour tous les professionnels de l'informatique qui cherchent à donner une impulsion significative à leur carrière. Les connaissances requises pour occuper un bon poste dans le secteur sont étendues, ce qui signifie qu'il existe de nombreuses possibilités de prospérer et même de diriger des équipes de développement.

Grâce à l'exhaustivité avec laquelle tous les contenus ont été développés de ce programme, le diplômé pourra diriger sa carrière vers le développement du *Front End Web*, mise en page, spécialiste de l'expérience client ou DevOps. Avec une vision à 360° de l'ensemble du processus de construction d'une application/web, l'informaticien sera capable d'aborder n'importe quel type de projet, apportant en outre un développement les dernières avancées dans tous les processus du cycle de vie d'un logiciel.

Une opportunité académique unique pour accéder à une connaissance qui allie la théorie informatique la plus récente à la pratique professionnelle de premier niveau, fournie par une équipe d'enseignants de la plus haute qualité. Son expérience à la tête de nombreux projets pertinents dans le domaine de la banque numérique ou des télécommunications font que le contenu didactique s'enrichisse, en apportant un grand nombre de cas réels et lectures complémentaires.

La grande flexibilité de cet enseignement est une autre de ses caractéristiques les plus remarquables. Il n'y a ni horaires prédéfinis ni cours présentiels, l'élève lui-même décidant quand, où et comment assumer toute la charge scolaire. La totalité du contenu de la salle de classe virtuelle est disponible en téléchargement, et peut être étudiée depuis n'importe quel appareil connecté à Internet.

Ce **Mastère Spécialisé en Programmation Front End Full Stack Developer** contient le programme académique le plus complet et le plus actuel du marché. Les principales caractéristiques sont les suivantes:

- ◆ Développement de cas pratiques présentés par des experts en programmation Front End Full Stack Developer
- ◆ Les contenus graphiques, schématiques et éminemment pratiques avec lesquels ils sont conçus fournissent des informations scientifiques et sanitaires essentielles à la pratique professionnelle
- ◆ Les exercices pratiques où le processus d'auto-évaluation peut être réalisé pour améliorer l'apprentissage
- ◆ Il met l'accent sur les méthodologies innovantes
- ◆ Cours théoriques, questions à l'expert et travail de réflexion individuel
- ◆ La possibilité d'accéder aux contenus depuis n'importe quel appareil fixe ou portable doté d'une connexion internet



Inscrivez-vous maintenant et ne laissez pas passer l'occasion qui vous mènera au zénith en leadership et développement des projets informatiques les plus ambitieux"

“

Vous atteindrez un niveau de spécialisation avancé et serez en mesure de construire n'importe quelle solution web requise avec une perspective d'Expérience Client moderne et adaptée au marché actuel"

Le corps enseignant du programme englobe des spécialistes réputés dans le domaine et qui apportent, à ce programme l'expérience de leur travail, ainsi que des spécialistes reconnus dans de grandes sociétés et des universités prestigieuses.

Grâce à son contenu multimédia développé avec les dernières technologies éducatives, les spécialistes bénéficieront d'un apprentissage situé et contextuel, ainsi, ils se formeront dans un environnement simulé qui leur permettra d'apprendre en immersion et de s'entraîner dans des situations réelles.

La conception de ce programme est axée sur l'Apprentissage Par les Problèmes, grâce auquel le professionnel doit essayer de résoudre les différentes situations de pratique professionnelle qui se présentent tout au long du Mastère Spécialisé. Pour ce faire, l'étudiant sera assisté d'un innovant système de vidéos interactives, créé par des experts reconnus.

Vous approfondirez les méthodologies agiles et comment elles peuvent être mises en œuvre dans le processus de développement, en augmentant vos capacités et compétences transversales.

Vous aurez accès à un matériel didactique complet, allant du langage de programmation Javascript aux outils tels que CSS, Angular et ReactJS.



02 Objectifs

L'objectif de ce Mastère Spécialisé en Programmation Front End Full Stack Developer, compte tenu des nombreuses options offertes par le développement, ne c'est autre chose que de fournir les connaissances et les techniques les plus avancées dans ce domaine. De cette façon, l'informaticien peut même commencer à développer ses propres projets ou stimuler son parcours professionnel avant la fin du diplôme, merci l'approche éminemment pratique de tous les contenus fournis.



“

Vous réaliserez vos objectifs professionnels les plus ambitieux grâce à l'approche distinctive de ce programme, qui vous emmènera partout les étapes du développement Front End Full Stack"



Objectifs généraux

- ◆ Générer des connaissances spécialisées sur aspects clés des programmation
- ◆ Encourager la pensée algorithmique
- ◆ Fournir les outils et les compétences nécessaires au développement
- ◆ Promouvoir l'adoption de méthodes souples pour la mise en oeuvre des projets
- ◆ Développer des connaissances spécialisées sur les bases du web
- ◆ Encourager l'utilisation des techniques et outils de développement *Front End* modernes
- ◆ Développement d'une conception Web pour traiter correctement la composition
- ◆ Évaluer les connaissances acquises





Objectifs spécifiques

Module 1. Développeur *Full Stack Developer*

- ◆ Développer des connaissances avancées en programmation
- ◆ Promouvoir l'utilisation de systèmes de contrôle de version et de plateformes d'hébergement de code
- ◆ Promouvoir l'utilisation des Méthodologies Agiles
- ◆ Approfondir les concepts clés et le fonctionnement d'internet
- ◆ Pour améliorer les compétences en matière de ligne de commande

Module 2. *Front-end* dans Programmation

- ◆ Identifier et comprendre la syntaxe correcte de HTML et CSS
- ◆ Explorer les différents éléments du HTML
- ◆ Déterminer l'approche de la conception adaptative
- ◆ Utiliser le formatage de la présentation des pages Web en appliquant des feuilles de style en cascade
- ◆ Incorporer le préprocesseur CSS
- ◆ Établir les avantages de l'utilisation d'un préprocesseur
- ◆ Générer des connaissances spécialisées sur les systèmes de conception
- ◆ Établir les critères d'utilisation du système de conception

Module 3. Le langage **JAVASCRIPT** appliqué au *Full Stack Developer*

- ◆ Établir les types de base et complexes offerts par JavaScript
- ◆ Analyser les différentes façons de programmer avec le langage et l'utiliser correctement dans chaque situation
- ◆ Mise à jour des connaissances vers les dernières versions
- ◆ Découvrir la programmation fonctionnelle
- ◆ Examiner la programmation asynchrone et ses caractéristiques

Module 4. Mise en page web appliquée au *Full Stack Developer*

- ◆ Évaluer un design web pour savoir comment le situer temporellement
- ◆ Examinez les principales règles CSS
- ◆ Présenter les différentes méthodologies CSS pour obtenir des designs *Responsive*
- ◆ Principes de base du développement CSS en cascade
- ◆ Identifier la technologie Bootstrap dans toute conception web
- ◆ Analyser les principes de Bootstrap
- ◆ Développer une maquette web à l'aide de Bootstrap
- ◆ Déterminer les principes de développement dans un projet SaSS

Module 5. Outils **JavaScript**. Bibliothèque **ReactJs**

- ◆ Déterminer les fonctionnalités de React
- ◆ Configurer un projet en utilisant Create React App
- ◆ Analyser le cycle de vie des composants dans React
- ◆ Acquérir des connaissances spécialisées sur les fonctionnalités modernes de React, telles que *Hooks* et *Context*
- ◆ Définir les états globaux en utilisant le *Context*
- ◆ Créer et rendre des listes et créer des formulaires avec React
- ◆ Implémenter la validation des champs dans les formulaires
- ◆ Composants et éléments de style
- ◆ Déboguer, tester et déployer des applications React

Module 6. Framework de JavaScript. Angular

- ◆ Développer une connaissance spécialisée de l'architecture de *Framework*
- ◆ Approfondir la méthodologie d'Angular
- ◆ Analyser le concept de composant
- ◆ Organiser le code correctement

Module 7. Programmation en langage NodeJS

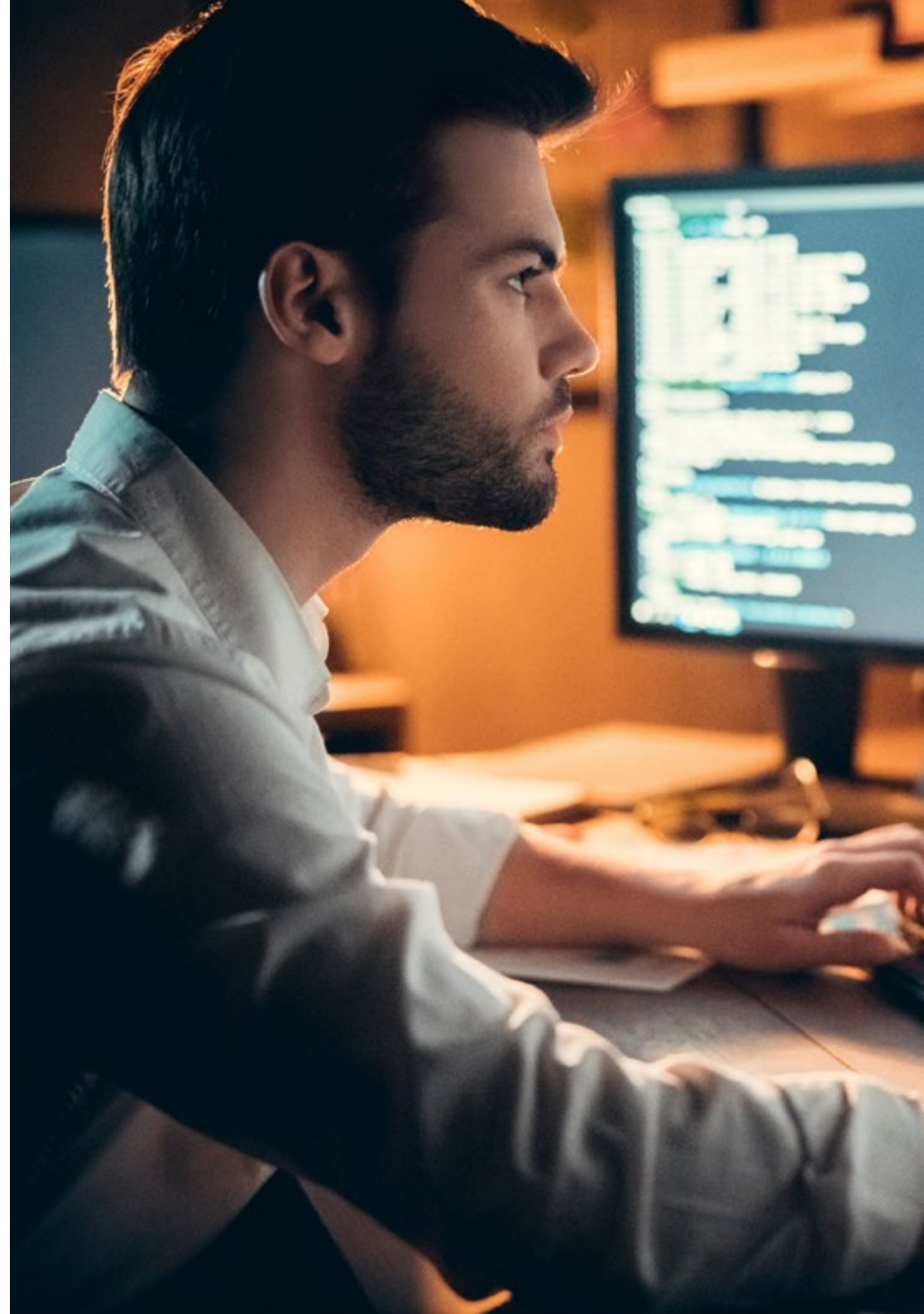
- ◆ Générer des connaissances spécialisées sur les types JavaScript et leurs opérateurs
- ◆ Analyser les meilleures façons de programmer avec le langage
- ◆ Mise à jour des connaissances vers les dernières versions
- ◆ Explorer la programmation fonctionnelle
- ◆ Développer la programmation asynchrone et sa motivation
- ◆ Acquérir la capacité d'exécuter une application avec NodeJS

Module 8. Bases de données pour développeurs *Full Stack*

- ◆ Détermination des raisons d'utiliser une base de données dans le développement d'applications
- ◆ Examiner les types de bases de données disponibles et leurs différences
- ◆ Développez une idée claire de l'utilisation de chaque type de base de données
- ◆ Analyser l'utilisation de la base de données dans les paradigmes de développement actuels

Module 9. UX CX. Expérience client

- ◆ Analyser l'importance de l'utilisateur aujourd'hui et approfondir la culture du retour
- ◆ Concrétiser les stratégies d'omnicanalité et de personnalisation basées sur les micro-interactions
- ◆ Étudier l'évolution de l'analyse web à l'analyse comportementale
- ◆ Déterminer comment l'Intelligence Artificielle a porté le CX au niveau supérieur



- ◆ Définir les techniques les plus importantes d'analyse de l'expérience Web, de la mobilité et de l'accessibilité
- ◆ Présentation de la méthodologie *Design Thinking* et du processus de création de l'expérience utilisateur
- ◆ Planifier des outils de prototypage et *WireFraming*, ainsi que *Frameworks* de développement de *Front-End*

Module 10. Intégration continue et déploiement d'applications

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

“

Vous améliorerez vos compétences progressivement, sur 10 modules élaborés à partir des connaissances les plus solides et de l'expérience éprouvée de tous les enseignants”

03

Compétences

Les compétences que doit développer un programmeur *Front End Full Stack* sont multiples, surtout dans un domaine aussi disputé et spécialisé qu'il l'est celui de l'informatique. C'est pourquoi l'agenda couvre les différents types de langage les plus courants dans ce domaine, ainsi que les outils et la philosophie de travail que doit suivre l'informaticien pour se démarquer avec une bonne distinction professionnelle. Cela est possible grâce à la nature multidisciplinaire de l'équipe enseignante elle-même, qui a mis en commun toutes ses connaissances dans divers domaines du développement du *Front End*.



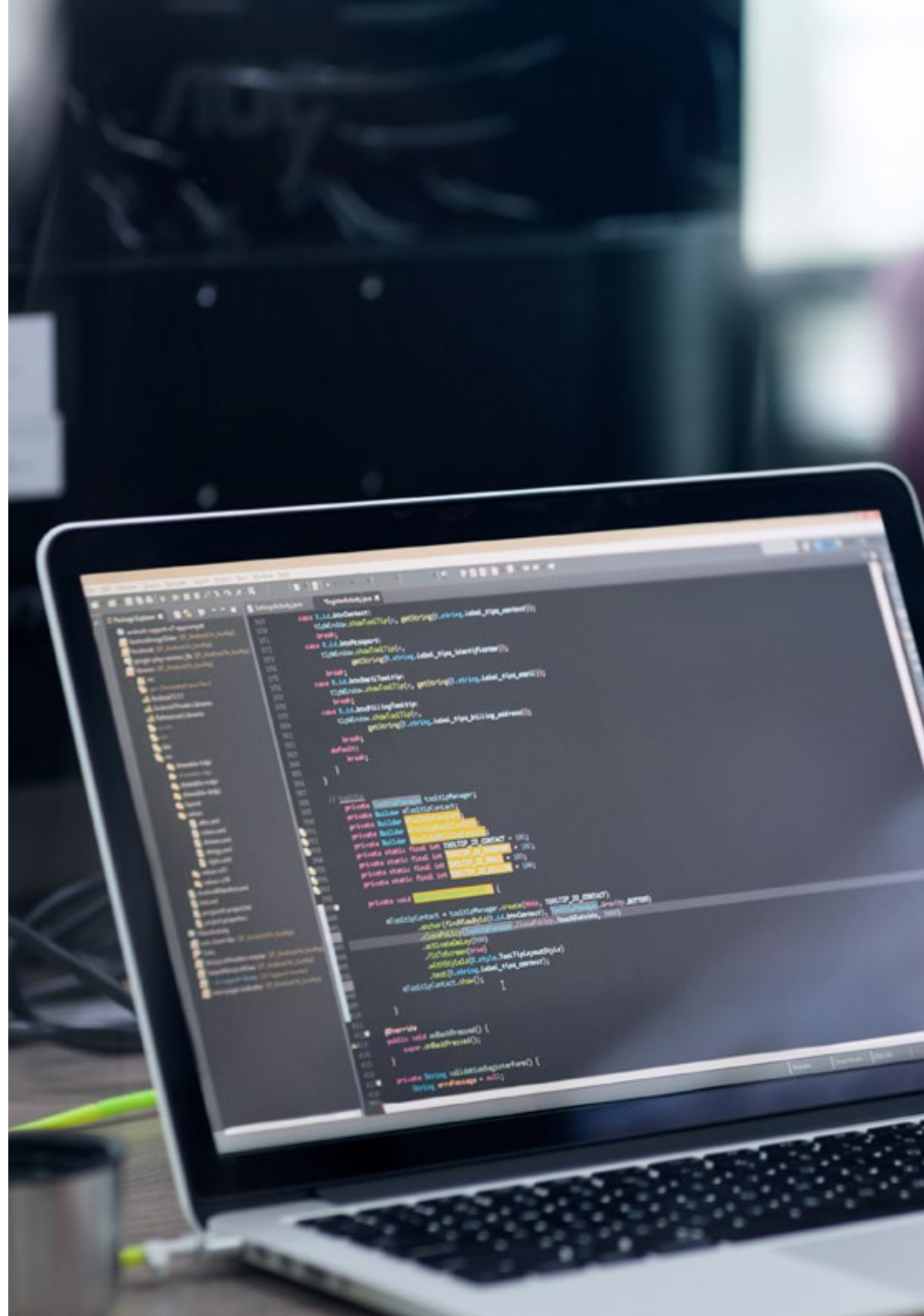


Vous donnerez un coup de pouce de grande qualité à votre CV en intégrant une multitude de compétences et de compétences requises dans les domaines les plus élevés du développement logiciel"



Compétences générales

- ◆ Reconnaître correctement la syntaxe des langages HTML et CSS
- ◆ Développer des critères de bonnes pratiques de développement web
- ◆ Générer des connaissances spécialisées sur le langage JavaScript
- ◆ Être capable de développer tout type d'application avec JavaScript
- ◆ Analyser la bibliothèque Bootstrap
- ◆ Exécuter des projets de composition avec SaSS (*Syntactically Awesome Stylesheets*)
- ◆ Identifier la syntaxe de React et comment la programmer
- ◆ Appliquer les bonnes pratiques au langage
- ◆ Examiner le processus de chargement et d'accès dans chacun des principaux types de base de données de votre région
- ◆ Évaluer les outils et les techniques les plus importants dans l'analyse CX et le "stack technologique" habituelle dans les entreprises





Compétences spécifiques

- ◆ Analyser les différentes structures de données
- ◆ Examiner les techniques de conception et d'interprétation des algorithmes
- ◆ Préparer l'environnement de développement
- ◆ Cloner un site Web
- ◆ Générer un site Web avec Bootstrap
- ◆ Compiler le code CSS avec SaSS
- ◆ Développer votre *Framework* CSS basé sur Bootstrap à l'aide de SaSS
- ◆ Générer un projet et le lancer
- ◆ Définir comment se connecter et charger/extraire des données à partir de différents types de bases de données
- ◆ Identifier les pratiques, les cas d'utilisation, les technologies et les outils de l'écosystème CI/CD, indispensables pour prendre en charge l'ensemble du processus



Tout au long du programme, vous développerez les compétences nécessaires pour réussir dans le développement du Front End, en renforçant non seulement vos propres connaissances, mais aussi vos compétences transversales"

04

Direction de la formation

TECH accorde une attention particulière à la sélection du personnel enseignant en charge de ses diplômes. L'exhaustivité des équipes garantit non seulement la qualité des contenus eux-mêmes, mais aussi l'implication d'une équipe engagée dans l'amélioration professionnelle de l'informatique. L'étudiant trouvera donc un matériel didactique adapté à ses critères les plus exigeants, avec toutes les aides et le matériel additionnel attendus de la plus grande institution académique en ligne du monde.



“

Vous pourrez consulter tous vos doutes directement avec le personnel enseignant, en obtenant un tutorat personnalisé et adapté à vos besoins”

Direction



M. Olalla Bonal, Martin

- ◆ *Actual Blockchain Technical Specialist en IBM SPGI*
- ◆ Technicien en Électronique Numérique
- ◆ Architecte *Blockchain*
- ◆ Architecte d'infrastructure dans le secteur bancaire
- ◆ Formation *Hyperledger Fabric* pour les entreprises
- ◆ Formation *Blockchain* en entreprise
- ◆ Gestion de projet et mise en œuvre de solutions en production
- ◆ Plus de 25 ans d'expérience dans le Secteurs Informatique

Professeurs

M. Calzada Martínez, Jesús

- ◆ Ingénieur de Software Senior en Devo
- ◆ Développeur Full *Full Stack* chez Blocknitive
- ◆ Responsable du *Front End* chez Infinia
- ◆ Développeur *Full Stack* chez Resem
- ◆ Développeur Java chez Hitec
- ◆ Diplômé en Ingénierie Informatique

M. Guerrero Díaz-Pintado, Arturo

- ◆ Ingénieur Technique Avant-vente via *Watson Customer Engagement portfolio* (solutions de *Marketing et Customer Experience*) en Espagne, au Portugal, en Grèce et en Israël chez IBM
- ◆ Ingénieur réseau R&D chez Telefonica
- ◆ Consultant en services professionnels travaillant avec des organisations de pointe en Europe, au Moyen-Orient et en Amérique latine, depuis IBM
- ◆ Diplômé en Génie Supérieur des Télécommunications de l'Université d'Alcalá et de *Danish Technical University*
- ◆ Des collaborations remarquables dans des universités et des établissements d'enseignement supérieur renommés dans des domaines liés à la technologie tels que l'Intelligence Artificielle, *Internet of Things*, *Cloud*, *Customer Experience* et *Digital Transformation*

M. Pintado San Claudio, Bruno

- ◆ Coordinateur du Développement chez IDavinci
- ◆ Développeur Java à la Bibliothèque Nationale d'Espagne
- ◆ Développeur de support et technicien réseau N1 à Sanitas
- ◆ Technicien d'appui système à la Mairie d'Alcobendas
- ◆ Technicien de communication N1 pour ADIF au Centre de Télécommunications d'Atocha
- ◆ Diplômé en Ingénierie Technique des Télécommunications avec spécialité en Systèmes Électroniques à l'Université Polytechnique de Madrid
- ◆ Diplômé en Ingénierie Électronique des Communications à l'Université Polytechnique de Madrid

M. Reyes Oliva, Luis

- ◆ Développeur et architecte cloud chez IBM
- ◆ Responsable technique client pour les comptes intégrés de BBVA chez IBM
- ◆ *Cloud Executive Selling* chez IBM
- ◆ Architecte Cloud et DevOps chez IBM
- ◆ Architecte logiciel client chez Telefónica
- ◆ Architecte de solutions techniques chez Rational
- ◆ *Software Engineering Manager* chez Borland
- ◆ Responsable du Développement Logiciel et de l'assurance qualité chez Altana Consulting
- ◆ Diplôme d'Ingénieur en Informatique de l'Universidad Pontificia de Salamanca, Madrid

M. Frias Favero, Pedro Luis

- ◆ CTO-Swearit Technologies
- ◆ COO-Key Identification
- ◆ Expert en *blockchain* et en applications décentralisées-Université de Alcalá
- ◆ Full Stack Developer-Ironhack
- ◆ Ingénieur Industriel diplômé de l'université de Yacambu

M. Gómez Rodríguez, Antonio

- ◆ Ingénieur de solutions Cloud chez Oracle
- ◆ Chef de Projet chez Sopra Group
- ◆ Chef de Projet à Everis
- ◆ Chef de Projet en Entreprise publique de Gestion des Programmes Culturels. Ministère de la Culture d'Andalousie
- ◆ Analyste des Systèmes d'Information. Sopra Group
- ◆ Diplôme d'Ingénieur Supérieur en Télécommunications de l'Université Polytechnique de Catalogne
- ◆ Diplôme en Technologies et Systèmes d'Information, Institut Catalan de Technologie
- ◆ *E-Business Master*, École de Commerce La Salle

05

Structure et contenu

Étant donné que l'effort et l'investissement que l'informaticien effectue dans l'option d'un programme de ces caractéristiques ne sont pas moindres, TECH mise sur une méthodologie pédagogique de premier plan, contrastée en efficacité et rentabilité. Le *Relearning* permet à l'élève d'acquérir les connaissances les plus importantes du programme tout au long du programme, évitant ainsi un temps d'étude précieux qui peut être mis à profit dans les multiples matériels complémentaires qui accompagnent chaque thème proposé.



“

Indaga dans les différents modules et thèmes de connaissances par le biais de vidéos résumés, de vidéos détaillées et de vidéos de motivation créées par les enseignants eux-mêmes”

Module 1. Développeur *Full Stack Developer*

- 1.1. Développement *Full Stack Developer* I. Programmation et langages
 - 1.1.1. Programmation
 - 1.1.2. Rôles de planification
 - 1.1.3. Langages et *Framework*
 - 1.1.4. Algorithmes
 - 1.1.5. Caractéristiques d'un algorithme
- 1.2. Développement *Full Stack Developer* II. Typologie
 - 1.2.1. Variables et constantes
 - 1.2.2. Types
 - 1.2.3. Opérateurs
 - 1.2.4. Déclarations
 - 1.2.5. Boucles
 - 1.2.6. Fonctions et objectifs
- 1.3. Structure de données dans le développement
 - 1.3.1. Types de structure linéaire
 - 1.3.2. Types de structures fonctionnelles
 - 1.3.3. Types de structures d'arborescence
- 1.4. Conception et interprétation d'algorithmes
 - 1.4.1. Parallélisme en développement. Diviser pour régner
 - 1.4.2. Algorithmes voraces
 - 1.4.3. Programmation dynamique
- 1.5. Environnement et outils de développement orienté *Full Stack Developer*
 - 1.5.1. Préparation de l'environnement pour Mac OS
 - 1.5.2. Préparation de l'environnement Linux
 - 1.5.3. Préparation de l'environnement Windows
- 1.6. Ligne de commande. Typologie et fonctionnement
 - 1.6.1. Le terminal
 - 1.6.2. Émulateurs
 - 1.6.3. Interprète de commande
 - 1.6.4. Premières commandes
 - 1.6.5. Navigation
 - 1.6.6. Gestion des fichiers et des dossiers à l'aide de l'interface de ligne de commande
 - 1.6.7. *Secure Shell*. SSH
 - 1.6.8. Commandes avancées
- 1.7. Git. Dépôt de logiciels
 - 1.7.1. Dépôt de logiciels Git
 - 1.7.2. Utilisation de Git
 - 1.7.3. Dépôt de logiciels
 - 1.7.4. Branches
 - 1.7.5. Cycle d'utilisation
 - 1.7.6. Commandes
- 1.8. Service d'hébergement de versions de code
 - 1.8.1. Service d'hébergement de versions de code
 - 1.8.2. Fournisseurs
 - 1.8.3. Référentiel
- 1.9. Internet
 - 1.9.1. Internet
 - 1.9.2. Protocoles utilisés dans les WWW
 - 1.9.3. Protocole HTTP
- 1.10. Méthodologies dans le développement *Full Stack*
 - 1.10.1. *Scrum*
 - 1.10.2. XP
 - 1.10.3. *Design Sprint*

Module 2. *Front-end* dans Programmation

- 2.1. Langage HTML
 - 2.1.1. Document HTML
 - 2.1.2. Élément *Head*
 - 2.1.3. Élément *Body*
 - 2.1.4. Texte
 - 2.1.5. Hyperliens
 - 2.1.6. Images
 - 2.1.7. *Primer Site*
- 2.2. Langage HTML *Layouts*
 - 2.2.1. Langage HTML Éléments
 - 2.2.2. *Layout* traditionnel
 - 2.2.3. *Layout* sémantique
- 2.3. Feuilles de style en cascade CSS (*Cascading Style Sheets*)
 - 2.3.1. Inclusion de CSS dans un document HTML
 - 2.3.2. Commentaires
 - 2.3.3. Sélecteurs
 - 2.3.4. Sélecteurs avancés
- 2.4. Propriétés CSS (*Cascading Style Sheets*)
 - 2.4.1. Couleur
 - 2.4.2. Texte
 - 2.4.3. Pseudo classes
 - 2.4.4. Transitions
 - 2.4.5. Animations
 - 2.4.6. Animation des éléments
 - 2.4.7. Animation avancée
- 2.5. Modèle de boîte
 - 2.5.1. Hauteur et largeur
 - 2.5.2. Marge
 - 2.5.3. Remplissages
- 2.6. Positionnement
 - 2.6.1. Positionnement statique
 - 2.6.2. Positionnement relatif
 - 2.6.3. Positionnement absolu
 - 2.6.4. Positionnement fixe
 - 2.6.5. *Flotteurs*
- 2.7. Conception adaptative
 - 2.7.1. *Viewport*
 - 2.7.2. *Media Queries*
 - 2.7.3. Unités CSS
 - 2.7.4. *Images*
 - 2.7.5. *Frameworks*
- 2.8. *Layout* moderne
 - 2.8.1. *Flex*
 - 2.8.2. *Grid*
 - 2.8.3. *Flex vs. Grid*
- 2.9. Préprocesseur
 - 2.9.1. Sass
 - 2.9.2. Variables
 - 2.9.3. Mixins
 - 2.9.4. Boucles
 - 2.9.5. Fonctions
- 2.10. Conception du Système
 - 2.10.1. Bootstrap
 - 2.10.2. Grille Bootstrap
 - 2.10.3. *Header* et *Footer* de notre site
 - 2.10.4. Formulaires
 - 2.10.5. *Cards*
 - 2.10.6. Modals

Module 3. Le langage JAVASCRIPT appliqué au *Full Stack Developer*

- 3.1. Types primitifs et opérateurs
 - 3.1.1. Langage JavaScript
 - 3.1.2. Les nombres et leurs opérateurs
 - 3.1.3. Chaînes de texte et leurs opérateurs
 - 3.1.4. Valeurs booléennes
 - 3.1.5. Conversion entre les types
- 3.2. Régulateurs de débit et structure
 - 3.2.1. Expressions et déclarations
 - 3.2.2. Variables et constantes
 - 3.2.3. Déclaration *If*
 - 3.2.4. Déclarations *For, While*
- 3.3. Fonctions
 - 3.3.1. Fonctions
 - 3.3.2. Paramètres
 - 3.3.3. Fonctions en tant que paramètres
 - 3.3.4. Champ d'application des variables
 - 3.3.5. *Scopes* imbriquées
 - 3.3.6. *Hoisting*
 - 3.3.7. *Fermetures*
 - 3.3.8. Récursion
- 3.4. Structures de données: objets
 - 3.4.1. Type *Object*
 - 3.4.2. Création d'objets
 - 3.4.3. Accès aux valeurs d'un objet
 - 3.4.4. Ajout ou suppression de propriétés
 - 3.4.5. Objets imbriqués
 - 3.4.6. *Destructuring* d'objets
 - 3.4.7. Méthodes du type *Object*
 - 3.4.8. *Spread Operator*
 - 3.4.9. Immutabilité
- 3.5. Structure des données: *Array*
 - 3.5.1. Structure des données. *Array*
 - 3.5.2. *Array*. Typologie
 - 3.5.3. *Arrays* emboîté
 - 3.5.4. Méthodes d'un *Array*
- 3.6. POO: *Prototype* et classe
 - 3.6.1. POO. Programmation orientée objet
 - 3.6.2. Prototypes
 - 3.6.3. Classe
 - 3.6.4. Données privées
 - 3.6.5. Sous-classes
 - 3.6.6. *Call* et *Apply*
- 3.7. Types de JavaScript
 - 3.7.1. *Set*
 - 3.7.2. *WeakSet*
 - 3.7.3. *Map*
 - 3.7.4. *WeakMap*
 - 3.7.5. Expressions régulières
- 3.8. Utilitaires JavaScript
 - 3.8.1. *Date*
 - 3.8.2. *Math*
 - 3.8.3. *Symbol*
 - 3.8.4. JSON
- 3.9. JavaScript dans le *Browser*
 - 3.9.1. Inclusion de JavaScript dans un site web
 - 3.9.2. DOM
 - 3.9.3. Événements
 - 3.9.4. *Storage* du navigateur
- 3.10. Programmation asynchrone
 - 3.10.1. Programmation asynchrone
 - 3.10.2. *Event Loop*
 - 3.10.3. *Callbacks*
 - 3.10.4. *Promises*
 - 3.10.5. *Async/Await*

Module 4. Mise en page web appliquée au *Full Stack Developer*

- 4.1. CSS et mise en page
 - 4.1.1. Mise en page avec des tableaux
 - 4.1.2. Conception des fluides
 - 4.1.3. L'ère du *Responsive*
 - 4.1.4. *Mobile First* vs. *Desktop First*
- 4.2. CSS et les règles de la conception web
 - 4.2.1. Sélecteurs
 - 4.2.2. Pseudo classes
 - 4.2.3. Pseudo-éléments
- 4.3. Mise en page avec CSS
 - 4.3.1. Règles du *Box Model*
 - 4.3.2. Typographie
 - 4.3.3. Couleurs
 - 4.3.4. Images
 - 4.3.5. Fonds
 - 4.3.6. Tableaux
 - 4.3.7. Formulaires
 - 4.3.8. Afficher et masquer des éléments
 - 4.3.9. Variables CSS
- 4.4. Design *Responsive* et design fluide
 - 4.4.1. Éléments flottants
 - 4.4.2. *Grid* CSS
 - 4.4.3. *Media Queries*
 - 4.4.4. *Flex Box*
- 4.5. La cascade CSS
 - 4.5.1. Priorité des règles CSS
 - 4.5.2. Règles dérogatoires
 - 4.5.3. Classes vs. Identifiants
- 4.6. SaSS
 - 4.6.1. Logiciel en tant que service (SaSS)
 - 4.6.2. Installation de SaSS
 - 4.6.3. Exécution et compilation de SaSS
 - 4.6.4. Structure d'un répertoire SaSS
- 4.7. Utilisation de SaSS
 - 4.7.1. Variables dans Sass
 - 4.7.2. Modulaire de notre projet
 - 4.7.3. Syntaxe SaSS
- 4.8. Logique SaSS
 - 4.8.1. Mixins
 - 4.8.2. Maps
 - 4.8.3. Fonctions et structures de contrôle
- 4.9. Composition avec Bootstrap
 - 4.9.1. Bootstrap
 - 4.9.2. *Layout* de Bootstrap
 - 4.9.3. Formulaires
 - 4.9.4. *Box Model* avec Bootstrap
 - 4.9.5. Couleurs et polices
 - 4.9.6. Liens et boutons
 - 4.9.7. Afficher et masquer les éléments avec Bootstrap
 - 4.9.8. *Flex Box* avec Bootstrap
 - 4.9.9. Composants
- 4.10. *Theming* Bootstrap
 - 4.10.1. Réécriture de Bootstrap avec SaSS (*Software as a Service*)
 - 4.10.2. Structure des fichiers
 - 4.10.3. Création de notre propre *Framework* CSS (*Cascading Style Sheets*)

Module 5. Outils de JavaScript. Bibliothèque ReactJs

- 5.1. Outil Javascript ReactJS
 - 5.1.1. L'outil ReactJS
 - 5.1.2. Create React App
 - 5.1.3. JavaScript *Syntax Extension*
- 5.2. Composants ReactJS
 - 5.2.1. Composants
 - 5.2.2. *Props*
 - 5.2.3. Renderings
- 5.3. Événements dans la bibliothèque ReactJS
 - 5.3.1. Gestion des événements
 - 5.3.2. Gestion des événements en ligne
 - 5.3.3. Événements dans la bibliothèque ReactJS
- 5.4. Configuration *Hooks* de ReactJS
 - 5.4.1. États d'un composant
 - 5.4.2. *Hook* d'état
 - 5.4.3. *Hook* à effet
 - 5.4.4. *Custom Hooks*
 - 5.4.5. Autres *Hooks*
- 5.5. *Component Context* dans ReactJS
 - 5.5.1. *Component Context* dans ReactJS
 - 5.5.2. Utilisation du *Context*
 - 5.5.3. Structure du *Context*
 - 5.5.4. *React. Create Context*
 - 5.5.5. *Context. Provider*
 - 5.5.6. *Class. Context Type*
 - 5.5.7. *Context. Consumer*
 - 5.5.8. *Context.displayName*
 - 5.5.9. Application pratique d'utilisation du *Context*
- 5.6. Routage dans ReactJs
 - 5.6.1. *Routeur*
 - 5.6.2. *React Router*
 - 5.6.3. Installation
 - 5.6.4. Routage de base
 - 5.6.5. Routage dynamique
 - 5.6.6. Composants primaires
 - 5.6.7. *React Router Hooks*
- 5.7. Utilisation de listes et de formulaires avec ReactJS
 - 5.7.1. Listes et boucles
 - 5.7.2. Formulaires et validations
 - 5.7.3. *React Hook Forms*
- 5.8. Utilisation des styles dans ReactJS
 - 5.8.1. Stylisé traditionnel
 - 5.8.2. Styles en ligne
 - 5.8.3. Agrégat de bibliothèque système de conception
- 5.9. Test en Javascript. Outils
 - 5.9.1. *Testing*
 - 5.9.2. *Jest JavaScript Testing Framework*
 - 5.9.3. *Visual Testing and Documentation*
- 5.10. Déploiement de code avec ReactJS
 - 5.10.1. *Hosting*
 - 5.10.2. Fournisseurs
 - 5.10.3. Préparation du projet
 - 5.10.4. Déploiement à Heroku



Module 6. *Framework* de JavaScript. Angular

- 6.1. Le *Framework* angular et son architecture
 - 6.1.1. Angular CLI
 - 6.1.2. Architecture
 - 6.1.3. *Workspace* et structure
 - 6.1.4. Environnement
- 6.2. Composants du *Framework* Angular
 - 6.2.1. Cycle de vie
 - 6.2.2. Encapsulation de la vue
 - 6.2.3. Interaction entre les composants
 - 6.2.4. Projection de contenu
- 6.3. Modèles de *Framework* Angular
 - 6.3.1. Interpolation de texte
 - 6.3.2. Déclarations
 - 6.3.3. *Property Binding*
 - 6.3.4. *Class, Style et Attribute Binding*
 - 6.3.5. *Event Binding et Two-Way Binding*
 - 6.3.6. *Pipes*
- 6.4. Directive de *Framework* Angular
 - 6.4.1. Stratégies Angular
 - 6.4.2. Stratégies d'attribut
 - 6.4.3. Stratégies de structure
- 6.5. Services d'injection de dépendances
 - 6.5.1. Services
 - 6.5.2. Injection de dépendances
 - 6.5.3. *Service Providers*

- 6.6. *Routing* et navigation
 - 6.6.1. Application avec *Routing*
 - 6.6.2. Routage de base
 - 6.6.3. Itinéraires imbriqués
 - 6.6.4. Paramètres
 - 6.6.5. Accès et autorisation
 - 6.6.6. *Lazy Loading* de modules
- 6.7. RxJS
 - 6.7.1. Observables
 - 6.7.2. Observers
 - 6.7.3. Abonnements
 - 6.7.4. Opérateurs
- 6.8. Formulaires et HTTP
 - 6.8.1. Formulaires réactifs
 - 6.8.2. Validation des champs
 - 6.8.3. Formulaires dynamiques
 - 6.8.4. Pétitions
 - 6.8.5. *Intercepteurs*
 - 6.8.6. Sécurité
- 6.9. Animations
 - 6.9.1. Transitions et *Triggers*
 - 6.9.2. Transitions de gamme
 - 6.9.3. Différences entre les transitions
- 6.10. *Testing* dans le *Framework* Angular
 - 6.10.1. Testage des services
 - 6.10.2. Testage des composants
 - 6.10.3. Testage des stratégies et des *pipes*

Module 7. Programmation en langage NodeJS

- 7.1. NodeJS et son architecture
 - 7.1.1. NPM et gestion des paquets
 - 7.1.2. Exécution d'un programme
 - 7.1.3. Modules
 - 7.1.4. Création d'un module
 - 7.1.5. *Loop* des événements
- 7.2. Servidor Backend, HTTP, *Express* et *Sockets*
 - 7.2.1. Module HTTP
 - 7.2.2. *Express*
 - 7.2.3. *Socket.io*
- 7.3. Bases de données et cache
 - 7.3.1. MongoDB
 - 7.3.2. Mongoose
 - 7.3.3. SQL
 - 7.3.4. *Sequelize*
 - 7.3.5. *Redis*
- 7.4. Système de fichiers et *Os*
 - 7.4.1. *File System Module*
 - 7.4.2. *Os Module*
 - 7.4.3. *Cluster Module*
- 7.5. Événements, *Buffers* et *Streams*
 - 7.5.1. Événements
 - 7.5.2. *Buffers*
 - 7.5.3. *Streams*
- 7.6. *Testing*
 - 7.6.1. *Jest*
 - 7.6.2. *Mocha*
 - 7.6.3. *TDD-Cucumber*

- 7.7. Architecture et bonnes pratiques
 - 7.7.1. DRY
 - 7.7.2. SOLID
 - 7.7.3. CRUD
 - 7.7.4. MVC
 - 7.7.5. Monolithes
 - 7.7.6. Microservices
 - 7.7.7. Architectures hexagonales
- 7.8. Typescript
 - 7.8.1. Types, interfaces et classes
 - 7.8.2. Fonctions et modules
 - 7.8.3. Génériques
 - 7.8.4. *Namespaces*
 - 7.8.5. Décorateurs
- 7.9. API REST
 - 7.9.1. Get
 - 7.9.2. Post
 - 7.9.3. Put
 - 7.9.4. Delete
 - 7.9.5. *Swagger*
 - 7.9.6. Construction d'une API REST avec Express
- 7.10. Création et conteneurisation d'une application avec NestJS
 - 7.10.1. Nest CLI
 - 7.10.2. Docker
 - 7.10.3. Construction d'une application

Module 8. Base de Données pour développeurs *Full Stack*

- 8.1. Bases de données pour développeurs *Full Stack*
 - 8.1.1. Base de données dans le développement d'applications
 - 8.1.2. Capacités de bases de données
 - 8.1.3. SQL (*Structured Query Language*)
- 8.2. Choix de la base de données
 - 8.2.1. Application ou service à considérer
 - 8.2.2. Catégories de base de données
 - 8.2.3. Panorama de base de données
- 8.3. Développement avec MySQL
 - 8.3.1. Développement avec MySQL
 - 8.3.2. Déploiement de modèle relationnel avec MySQL
 - 8.3.3. Connexion à MySQL
- 8.4. Développement avec Oracle Database
 - 8.4.1. Développement avec Oracle DB
 - 8.4.2. Déploiement du modèle
 - 8.4.3. Connexion à Oracle Database
- 8.5. Développement avec Oracle SQL Server
 - 8.5.1. Oracle SQL Server
 - 8.5.2. Déploiement du modèle
 - 8.5.3. Connexion à SQL Server
- 8.6. Développement avec NoSQL
 - 8.6.1. Comparaison avec les bases de données SQL
 - 8.6.2. Création de base de données dans MongoDB
 - 8.6.3. Connexion à MongoDB
- 8.7. Développement avec graphes
 - 8.7.1. Développement avec graphes
 - 8.7.2. Création de base de données avec Neo4j
 - 8.7.3. Connexion à Neo4j

- 8.8. Développement avec base de données clé-valeur
 - 8.8.1. Développement avec base de données k-v
 - 8.8.2. Création d'une base de données Redis
 - 8.8.3. Connexion Redis
- 8.9. Bases de données avec autres types de données
 - 8.9.1. *Elastic Search*
 - 8.9.2. *Inmemory Database*
 - 8.9.3. Développement des données spatiales
- 8.10. Bases de données. Aspects avancés
 - 8.10.1. Bases de données dans les développements cloud natif
 - 8.10.2. Bases de données d'architecture microservices
 - 8.10.3. CI/CD et bases de données

Module 9. UX CX. Expérience client

- 9.1. *Customer Experience*
 - 9.1.1. *Customer Experience (CX)*
 - 9.1.2. Nouveaux besoins de consommation
 - 9.1.3. *Feedback* dans *Customer Experience*
- 9.2. Technologies novatrices
 - 9.2.1. Machines à penser
 - 9.2.2. Nouvelles façons de partager des informations
 - 9.2.3. Mesure de ce qui ne peut pas être mesuré
- 9.3. Canaux d'interaction utilisateur
 - 9.3.1. Analyse client
 - 9.3.2. Personnalisation
 - 9.3.3. Plusieurs canaux d'interaction utilisateur
- 9.4. Analyse utilisateur
 - 9.4.1. Structure du site web
 - 9.4.2. Analyse utilisateur
 - 9.4.3. Analyse utilisateur avancée
- 9.5. Nielsen et son impact sur la CX
 - 9.5.1. Nielsen et son impact sur la CX
 - 9.5.2. Techniques de *User Testing*
- 9.6. Outils de *Customer Experience*
 - 9.6.1. Outils avancés
 - 9.6.2. Mobilité
 - 9.6.3. Accessibilité
- 9.7. Nouvelles méthodes
 - 9.7.1. Le défi de l'utilisateur
 - 9.7.2. Processus UX
 - 9.7.3. Recherche utilisateur
- 9.8. Communication d'une conception
 - 9.8.1. *Wireframing*
 - 9.8.2. Outils de communication d'une conception
 - 9.8.3. Outils de communication avancés d'une conception
- 9.9. Conception d'UI
 - 9.9.1. Conception d'UI
 - 9.9.2. Interfaces web et mobiles
 - 9.9.3. Composants web et mobiles
- 9.10. Élaboration d'une CX
 - 9.10.1. Élaboration d'une CX
 - 9.10.2. Conception de nouvelles expériences
 - 9.10.3. Interfaces

Module 10. Intégration continue et déploiement d'applications

- 10.1. Intégration continue et déploiement continu: CI/CD
 - 10.1.1. Utilisation de l'intégration continue et du déploiement continu (CI/CD)
 - 10.1.2. Différences entre intégration continue et déploiement continu (CI/CD)
 - 10.1.3. Intégration continue et déploiement continu. Avantages du CI/CD
- 10.2. Les nouveaux paradigmes de développement
 - 10.2.1. De Waterfall à DevOps
 - 10.2.2. Guide de style: les 12 facteurs
 - 10.2.3. Cloud Native, microservices et *Serverless*
- 10.3. DevOps, au-delà de CI/CD
 - 10.3.1. DevOps
 - 10.3.2. DevOps. *Continuous Everything*
 - 10.3.3. DevOps vs SRE
- 10.4. Technologie de conteneurs I-Docker
 - 10.4.1. Les conteneurs. Contributions
 - 10.4.2. Docker. Architecture
 - 10.4.3. Processus de déploiement avec Docker
- 10.5. Technologie de conteneurs II-Kubernetes
 - 10.5.1. Orchestration
 - 10.5.2. Kubernetes
 - 10.5.3. L'écosystème de Kubernetes
- 10.6. Configuration de l'infrastructure avec GitOps
 - 10.6.1. Infrastructure immuable
 - 10.6.2. GitOps
 - 10.6.3. Outils GitOps
- 10.7. Pipelines et automatisation. Cas d'Utilisation de CI/CD
 - 10.7.1. Intégration continue
 - 10.7.2. Déploiement et livraison continue
 - 10.7.3. Validations automatiques
 - 10.7.4. Bonnes pratiques en matière de CI/CD
- 10.8. CI/CD avec Jenkins. Référents
 - 10.8.1. CI/CD avec Jenkins
 - 10.8.2. Pipelines Jenkins
 - 10.8.3. Bonnes pratiques avec Jenkins
- 10.9. Écosystème CI/CD
 - 10.9.1. Organisation des écosystèmes
 - 10.9.2. Outils avancés
 - 10.9.3. Dagger. Le futur
- 10.10. Phases finales du cycle du logiciel orienté CI/CD
 - 10.10.1. Application de l'IA au processus CI/CD
 - 10.10.2. DevSecOps
 - 10.10.3. *Chaos Engineering*



Vous disposerez des moyens éducatifs les plus modernes, avec un accès libre à la salle de classe virtuelle 24 heures par jour"

06

Méthodologie

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

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



“

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

Étude de Cas pour mettre en contexte tout le contenu

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

“

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



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



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

Une méthode d'apprentissage innovante et différente

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

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

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

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

Relearning Methodology

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

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

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

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

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



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

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

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

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



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



Support d'étude

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

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



Cours magistraux

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

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



Pratiques en compétences et aptitudes

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



Lectures complémentaires

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





Case studies

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



Résumés interactifs

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



Testing & Retesting

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



07 Diplôme

Le Mastère Spécialisé en Programmation Front End Full Stack Developer 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.



“

Réussissez ce programme avec succès et recevez votre diplôme universitaire sans déplacements ni formalités fastidieuses”

Ce **Mastère Spécialisé en Programmation Front End Full Stack Developer** 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 Programmation Front End Full Stack Developer**
N.º 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.



Mastère Spécialisé
Programmation Front
End Full Stack Developer

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

Mastère Spécialisé

Programmation Front End Full Stack Developer