



Mastère Hybride

Programmation Front End
Full Stack Developer

Modalité: Hybride (en ligne + Stage Pratique)

Durée: 12 mois

Diplôme: TECH Université Technologique

Accès au site web: www.techtute.com/fr/informatique/mastere-hybride/mastere-hybride-programmation-front-end-full-stack-developer

Sommaire

01

Présentation

Page 4

02

Pourquoi suivre ce
Mastère Hybride?

Page 8

03

Objectifs

Page 12

04

Compétences

Page 18

05

Direction de la formation

Page 22

06

Structure et contenu

Page 26

07

Stage Pratique

Page 38

08

Où puis-je effectuer
mon Stage Pratique?

Page 44

09

Méthodologie

Page 48

10

Diplôme

Page 56

01

Présentation

Les entreprises recherchent des développeurs qui ont non seulement la capacité de créer des interfaces utilisateur attrayantes et fonctionnelles, mais qui maîtrisent également le développement de serveurs et de bases de données. Avec l'essor de technologies telles que React, Node.js et l'importance croissante de l'expérience utilisateur (UX), être capable d'intégrer plusieurs couches de développement web est très apprécié. En effet, avec l'évolution constante des méthodologies agiles et de l'automatisation des Processus, les *Full Stack Developers* doivent toujours être à jour. Dans ce contexte, TECH a développé ce programme complet, qui se divise en deux parties, une première totalement en ligne, basée sur la méthodologie innovante *Relearning*, et un séjour pratique de 3 semaines dans une entreprise de renom.



“

Grâce à ce Mastère Hybride, vous maîtriserez les techniques de programmation, tout en acquérant des compétences en matière de gestion de projet, d'expérience utilisateur et de déploiement continu"

La demande de professionnels en Programmation Front End Full Stack continue d'augmenter, stimulée par l'accélération de la transformation numérique dans de nombreux secteurs. Avec la prolifération des applications web et mobiles, les entreprises recherchent des développeurs capables de gérer à la fois le *front-end* et le *back-end*.

C'est ainsi qu'est né ce Mastère Hybride, qui se déclinera en deux parties distinctes. Tout d'abord, une partie théorique qui fournira une vue d'ensemble des technologies et des compétences nécessaires au développement complet d'applications web, y compris un accent sur la Programmation *front-end*, où les informaticiens se concentreront sur les techniques et les meilleures pratiques pour créer des interfaces utilisateur efficaces et attrayantes.

Le programme d'études se penchera également sur le langage JavaScript appliqué à *Full Stack Developer*, crucial pour tout développeur moderne, ainsi que sur la mise en page web appliquée, où les professionnels structureront et styliseront les pages web d'une manière professionnelle. En outre, les outils JavaScript et les bibliothèques spécifiques, telles que React.js, seront indispensables pour le développement d'applications dynamiques et évolutives.

De même, la programmation dans le langage Node.js sera couverte en profondeur, ainsi que les bases de données pour les Développeurs *Full Stack*, y compris la logique du serveur et la gestion des bases de données. Il abordera également l'importance de l'expérience utilisateur, UX CX, et se terminera par l'intégration continue et le déploiement d'applications. Enfin, la deuxième partie de la formation consistera en un stage pratique approfondi de 3 semaines dans l'une des meilleures entreprises informatiques, soigneusement sélectionnée pour faire passer la carrière du diplômé au niveau supérieur.

TECH a ainsi conçu un programme entièrement en ligne et totalement adaptable, qui ne nécessitera qu'un appareil électronique doté d'une connexion Internet pour accéder au matériel pédagogique. Il sera également basé sur la méthodologie révolutionnaire *Relearning*, pionnière dans cette université, qui consiste à réitérer les concepts clés pour une assimilation optimale et organique du contenu.

Ce **Mastère Hybride en Programmation Front End Full Stack Developer** contient le programme le plus complet et le plus actualisé du marché. Ses caractéristiques sont les suivantes:

- ♦ Développement de plus de 100 études de cas présentées par des experts professionnels en Programmation Front End Full Stack Developer et des professeurs d'université ayant une grande expérience dans ce domaine
- ♦ Les contenus graphiques, schématiques et éminemment pratiques avec lesquels elles sont conçues fournissent des informations essentielles sur les outils et les techniques indispensables à la pratique professionnelle
- ♦ Tout cela sera complété par des cours théoriques, des questions à l'expert, des forums de discussion sur des sujets controversés et un travail de réflexion individuel
- ♦ Disponibilité des contenus à partir de tout appareil fixe ou portable doté d'une connexion internet
- ♦ En outre, vous pourrez effectuer un stage dans l'une des meilleures entreprises du secteur



Vous vous plongerez dans la création d'interfaces utilisateur, avec des techniques HTML, CSS et JavaScript avancées pour concevoir des expériences utilisateur intuitives et réactives. Qu'attendez-vous pour vous inscrire?"

“

Vous vous concentrerez sur le langage JavaScript appliqué au Full Stack Developer, en maîtrisant JavaScript, l'épine dorsale de la plupart des applications web modernes. Avec toutes les garanties de qualité!"

Dans cette proposition de Mastère, de nature professionnalisante et de modalité hybride, le programme est destiné à la mise à jour des professionnels de l'informatique qui exercent leurs fonctions dans des centres spécialisés en développement de software, et qui requièrent un haut niveau de qualification. Le contenu est basé sur les dernières données scientifiques, et orientés de manière didactique pour intégrer les connaissances théoriques dans la pratique informatique, et les éléments théoriques et pratiques faciliteront la mise à jour des connaissances et permettront la prise de décision stratégique éclairée.

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

Effectuez une pratique clinique de 3 semaines dans une entreprise prestigieuse et acquérez toutes les connaissances nécessaires à votre développement personnel et professionnel.

Vous analyserez les principes de la conception centrée sur l'utilisateur, en améliorant l'expérience du client, grâce aux meilleurs matériels pédagogiques sur le marché universitaire, à la pointe de la technologie et de l'éducation.



02

Pourquoi suivre ce Mastère Hybride?

Suivre ce Mastère Hybride est une décision stratégique pour ceux qui cherchent à exceller dans le domaine compétitif du développement web. Ainsi, ce programme offrira une combinaison idéale de flexibilité et de structure, permettant aux informaticiens de concilier leurs études avec d'autres responsabilités, tout en accédant à une formation de haute qualité. En outre, le parcours comprend un stage complet de 3 semaines dans une entreprise informatique prestigieuse, où les professionnels seront soutenus par une équipe d'experts dans le domaine. Cette qualification améliorera non seulement l'employabilité, mais permettra également aux diplômés de relever n'importe quel défi technologique.



“

Vous aurez la possibilité d'apprendre grâce à une méthodologie hybride, qui comprend des cours en ligne et un stage pratique innovant, facilitant ainsi une expérience d'apprentissage plus dynamique et plus pratique”

1. Actualisation des technologies les plus récentes

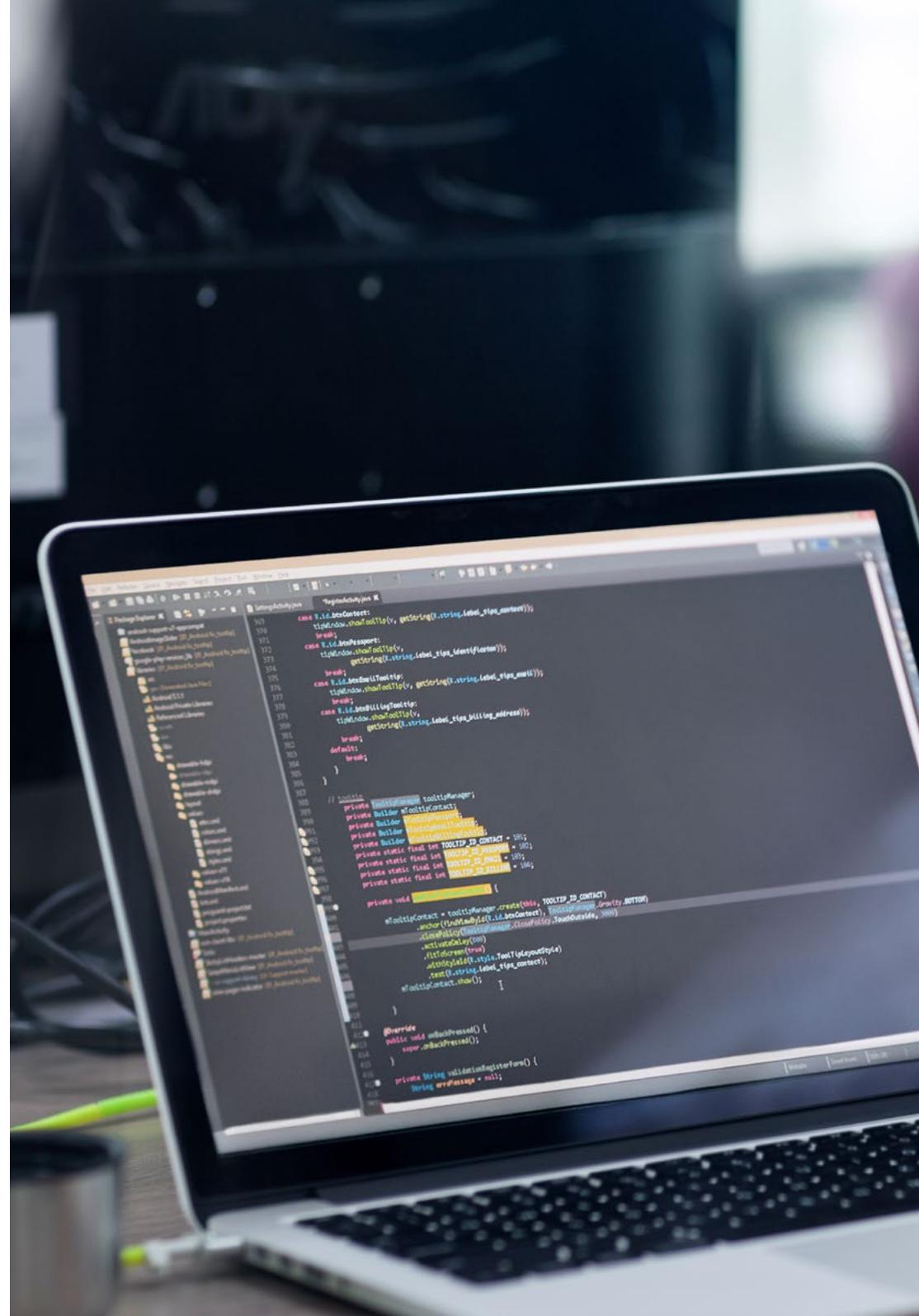
Les dernières technologies en matière de Programmation Front End Full Stack Developer se caractérisent par l'intégration d'outils et de *frameworks* avancés qui facilitent le développement d'applications web plus rapides, plus sûres et plus évolutives. Sur le *front-end*, des *frameworks* tels que React et Angular continuent d'évoluer, offrant des améliorations en termes de performances et de fonctionnalités. En ce qui concerne le *back-end*, Node.js continue d'être un outil fondamental. Ces innovations permettent non seulement d'améliorer la productivité des développeurs, mais aussi d'élever le niveau des applications web en termes de performances et de convivialité.

2. Exploiter l'expertise des meilleurs spécialistes

L'importante équipe de professionnels qui accompagnera le spécialiste tout au long de la période pratique est une prestation de premier ordre et une garantie de mise à jour sans précédent. Avec un tuteur spécialement désigné, l'informaticien pourra voir de vrais patients dans un environnement de pointe, ce qui lui permettra d'intégrer dans sa pratique quotidienne les procédures et les outils les plus efficaces en matière de Programmation Front End Full Stack Developer.

3. Accéder à des environnements professionnels de premier ordre

TECH sélectionne soigneusement tous les centres disponibles pour la Formation Pratique. Grâce à cela, le spécialiste aura un accès garanti à un environnement clinique de prestige dans le domaine de la Programmation Front End Full Stack Developer. Elle pourra ainsi expérimenter le travail quotidien dans un domaine exigeant, rigoureux et exhaustif, en appliquant toujours les thèses et postulats scientifiques les plus récents dans sa méthodologie de travail.



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

Le marché académique est miné par des programmes d'enseignement mal adaptés au travail quotidien du spécialiste et qui nécessitent de longues heures d'enseignement, souvent incompatibles avec la vie personnelle et professionnelle. TECH propose donc un nouveau modèle d'apprentissage, 100% pratique, qui lui permet de se familiariser avec les dernières procédures dans le domaine de la Programmation Front End Full Stack Developer et, mieux encore, de les mettre en pratique professionnelle en seulement 3 semaines.

5. Ouvrir la porte à de nouvelles opportunités

Les développeurs *Full Stack*, capables de gérer à la fois le *front-end* et le *back-end* d'une application, sont très appréciés pour leur polyvalence et leur efficacité dans la création de solutions de bout en bout. Ce domaine complet leur permet de travailler sur une variété de projets, des startups innovantes aux grandes entreprises technologiques, et de s'adapter à différents rôles, que ce soit en tant que développeur, architecte de software ou responsable technique. Ainsi, avec l'accent mis de plus en plus sur la numérisation et la transformation numérique, les développeurs *Full Stack* sont dans une position privilégiée pour tirer parti de ces opportunités et faire progresser leur carrière.

“

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

03

Objectifs

Ce programme universitaire cherchera à doter les informaticiens d'une connaissance approfondie et pratique des technologies les plus avancées en matière de programmation web, telles que HTML, CSS, JavaScript, React, Angular, Node.js et les bases de données relationnelles et non relationnelles. En outre, des compétences en méthodologies agiles, en gestion de projet et en outils d'intégration et de déploiement continus seront développées, garantissant que les diplômés peuvent s'adapter aux demandes dynamiques de l'industrie technologique. Un autre objectif clé sera d'inculquer une approche centrée sur l'utilisateur, à travers les principes UX et CX pour créer des expériences numériques exceptionnelles.



“

À la fin de ce Mastère Hybride, vous serez prêt à relever avec succès les défis du marché du travail, en vous distinguant par votre polyvalence, votre capacité d'innovation et vos compétences techniques avancées”



Objectif général

- L'objectif général du Mastère Hybride en Programmation Front End Full Stack Developer est de fournir des connaissances spécialisées dans les aspects clés de la programmation, en se concentrant sur les fondamentaux essentiels du web, y compris la syntaxe correcte des langages HTML et CSS. En outre, les professionnels seront formés pour développer tout type d'application à l'aide de JavaScript, y compris l'utilisation de la bibliothèque Bootstrap et la mise en œuvre de projets de mise en page avec SaSS. Ils développeront également des compétences avancées dans l'utilisation de *frameworks* modernes, tels que React et Angular, en explorant le potentiel du langage NodeJS.

“

Vous gérerez des bases de données relationnelles et non relationnelles, essentielles pour le stockage et la manipulation des données, aux mains de la meilleure université numérique du monde, selon Forbes: TECH”





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* en 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 du *framework*
- ♦ Approfondir la méthodologie d'angulaire
- ♦ 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 Node.js

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

04

Compétences

Les compétences clés de ce diplôme académique incluront la capacité à concevoir et à développer des applications web complètes, en utilisant des technologies *front-end* et *back-end* avancées, telles que HTML, CSS, JavaScript, React, Angular et Node.js. Les informaticiens acquerront également des compétences dans la mise en œuvre des pratiques UX/UI pour améliorer l'expérience utilisateur et l'intégration des bases de données pour une gestion efficace des données dans les applications. En outre, l'accent sera mis sur l'utilisation de méthodologies agiles et d'outils CI/CD pour assurer la livraison continue et l'amélioration constante des produits numériques.

```
...ings.js  
...js  
.js command.js  
.js editor.js  
.js fileManager.js  
.js main.js  
... readme.rst  
.js sequences.js
```

```
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66
```

```
self  
input.on  
//escap  
if (e.  
// th  
self.  
retur  
}  
if (e.  
e.st  
e.pr  
self  
self  
retur  
}  
//up/d  
if (e.  
e.pr  
e.st  
if (c  
//  
if
```

```
searchHistory;  
= this;  
("keydown", function(e) {  
  if (e.keyCode == 27) {  
    // this is a new line  
    .deactivate(true);  
    // this is a modified line  
  }  
});
```

```
keyCode == 13) {  
  .stopImmediatePropagation();  
  .eventDefault();  
  .search();  
  .deactivate();  
  return;  
}
```

```
down  
keyCode == 38 || e.keyCode == 40) {  
  .eventDefault();  
  .stopImmediatePropagation();  
  .show();  
}
```

“

Vous vous plongerez dans les pratiques DevOps, garantissant le déploiement et la maintenance efficaces et sécurisés des applications, grâce à une vaste bibliothèque de ressources multimédias”



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 la "pile 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 un *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

05

Direction de la formation

L'équipe enseignante derrière le Mastère Hybride en Programmation Front End Full Stack Developer sont des professionnels experts avec une vaste expérience dans le domaine du développement web. En effet, ils s'engagent à fournir un enseignement de qualité et actualisé, basé sur leur connaissance approfondie des technologies telles que HTML, CSS, JavaScript, React, Angular, Node.js, et diverses bases de données. En plus de leur expertise technique, ils sont compétents dans l'application des méthodologies agiles et de la gestion de projet, garantissant que les diplômés acquièrent à la fois des compétences pratiques et une solide compréhension des principes fondamentaux du développement *Full Stack*.



“

Le dévouement du corps professoral ne consiste pas seulement à transmettre des connaissances théoriques et pratiques, mais aussi à vous guider vers l'excellence professionnelle, en vous préparant à exceller dans l'environnement de travail"

Direction



M. Olalla Bonal, Martín

- ◆ Responsable de la Pratique de *Blockchain* chez EY
- ◆ Spécialiste Technique Client *Blockchain* pour IBM
- ◆ Directeur de l'Architecture de Blocknitive
- ◆ Coordinateur de l'Équipe Bases de Données Distribuées non Relationnelles pour wedoIT, Filiale d'IBM
- ◆ Architecte d'Infrastructure chez Bankia
- ◆ Chef du Département Mise en Page chez T-Systems
- ◆ Coordinateur de Département pour Bing Data España SL

Professeurs

M. Calzada Martínez, Jesús

- ◆ Ingénieur de Software Senior en Devo
- ◆ Développeur Full *Full Stack* chez Blocknitive
- ◆ Responsable de *Frontend* à Infinia
- ◆ Développeur *Full Stack* chez Resem
- ◆ Développeur Java chez Hitec
- ◆ Diplôme en Ingénierie Informatique de l'Université Complutense de Madrid

M. Frias Favero, Pedro Luis

- ◆ Architecte Principal de la Blockchain chez EY
- ◆ Co-fondateur et Directeur Technique de Swear IT Technologies
- ◆ Directeur du Support Informatique IT pour Key Business One. Mexique, Colombie et Espagne
- ◆ Diplôme d'Ingénieur Industriel de l'Université Yacambú
- ◆ Expert en Blockchain et Applications Décentralisées à l'Université d'Alcalá

M. Reyes Oliva, Luis

- ♦ Architecte de *Software et Cloud*
- ♦ Développeur et Architecte *Cloud* chez IBM
- ♦ Responsable Technique Client pour les comptes intégrés de BBVA chez IBM
- ♦ Architecte *Cloud* et *DevOps* chez IBM
- ♦ Architecte de *software* pour les clients chez Telefónica
- ♦ Architecte de solutions techniques chez Rational
- ♦ *Software Engineering Manager* chez Borland
- ♦ *Project Manager* à Altana Consulting
- ♦ Licence en Ingénierie Informatique de l'Université Pontificale de Salamanque

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
- ♦ Diplôme en Ingénierie Technique des Télécommunications avec Spécialité en Systèmes Électroniques à l'Université Polytechnique de Valence
- ♦ Diplôme en Ingénierie Électronique des Communications à l'Université Polytechnique de Madrid

M. Gómez Rodríguez, Antonio

- ♦ Ingénieur Principal de Solutions Cloud chez Oracle
- ♦ Co-organisateur de Malaga Developer Meetup
- ♦ Consultant Spécialisé pour Sopra Group et Everis
- ♦ Chef d'équipe chez System Dynamics
- ♦ Développeur de Logiciels chez SGO Software
- ♦ Master en E-Business de l'École de Commerce de La Salle
- ♦ Diplôme en Technologies et Systèmes d'Information de l'Institut Catalan de Technologie
- ♦ Diplôme d'Ingénieur Supérieur en Télécommunications de l'Université Polytechnique de Catalogne

M. Guerrero Díaz-Pintado, Arturo

- ♦ Directeur de l'Expérience Client chez IBM
- ♦ Ingénieur Technique de Pré-vente pour *Watson Customer Engagement* portefeuille
- ♦ Ingénieur réseau R&D chez Telefonica
- ♦ Licence en Ingénierie Supérieure des Télécommunications de l'Université d'Alcalá et de la *Danish Technical University*

06

Structure et contenu

Le contenu du programme ira des bases essentielles, telles que HTML, CSS et JavaScript, aux technologies avancées, telles que React, Angular et Node.js, pour le développement *front-end* et *back-end* d'applications web. Ils apprendront la mise en page web, la conception d'interfaces et la mise en œuvre de pratiques UX/UI pour optimiser l'expérience utilisateur. En outre, ils apprendront les techniques d'intégration continue et de déploiement d'applications (CI/CD), ainsi que la gestion efficace des bases de données et l'application des méthodologies agiles pour la gestion de projet.



“

Le Mastère Hybride en Programmation Front End Full Stack Developer couvrira un large éventail de contenus, conçus pour vous fournir une formation complète en développement web moderne”

Module 1. Développement *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. Algorithme
 - 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. Système de conception
 - 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. Structure des 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 Classes
 - 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. Rendering
- 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* angulaire et son architecture
 - 6.1.1. Angular CLI
 - 6.1.2. Architecture
 - 6.1.3. *Workspace* et structure
 - 6.1.4. Entourage
- 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 composantes
 - 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 native*
 - 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*

“

Cette approche complète et pratique vous préparera à relever avec succès les défis du marché du travail actuel, en vous dotant des compétences nécessaires pour développer et maintenir des applications web sophistiquées et fonctionnelles”

07

Stage Pratique

Après avoir passé la période théorique en ligne, le programme comprendra une période de 3 semaines de formation pratique dans une entreprise de référence spécialisée dans la programmation et le développement web. Le diplômé disposera ainsi du soutien d'un tuteur personnel, qui l'accompagnera tout au long du processus et consacrera 100% de son temps à la préparation et au déroulement du stage.



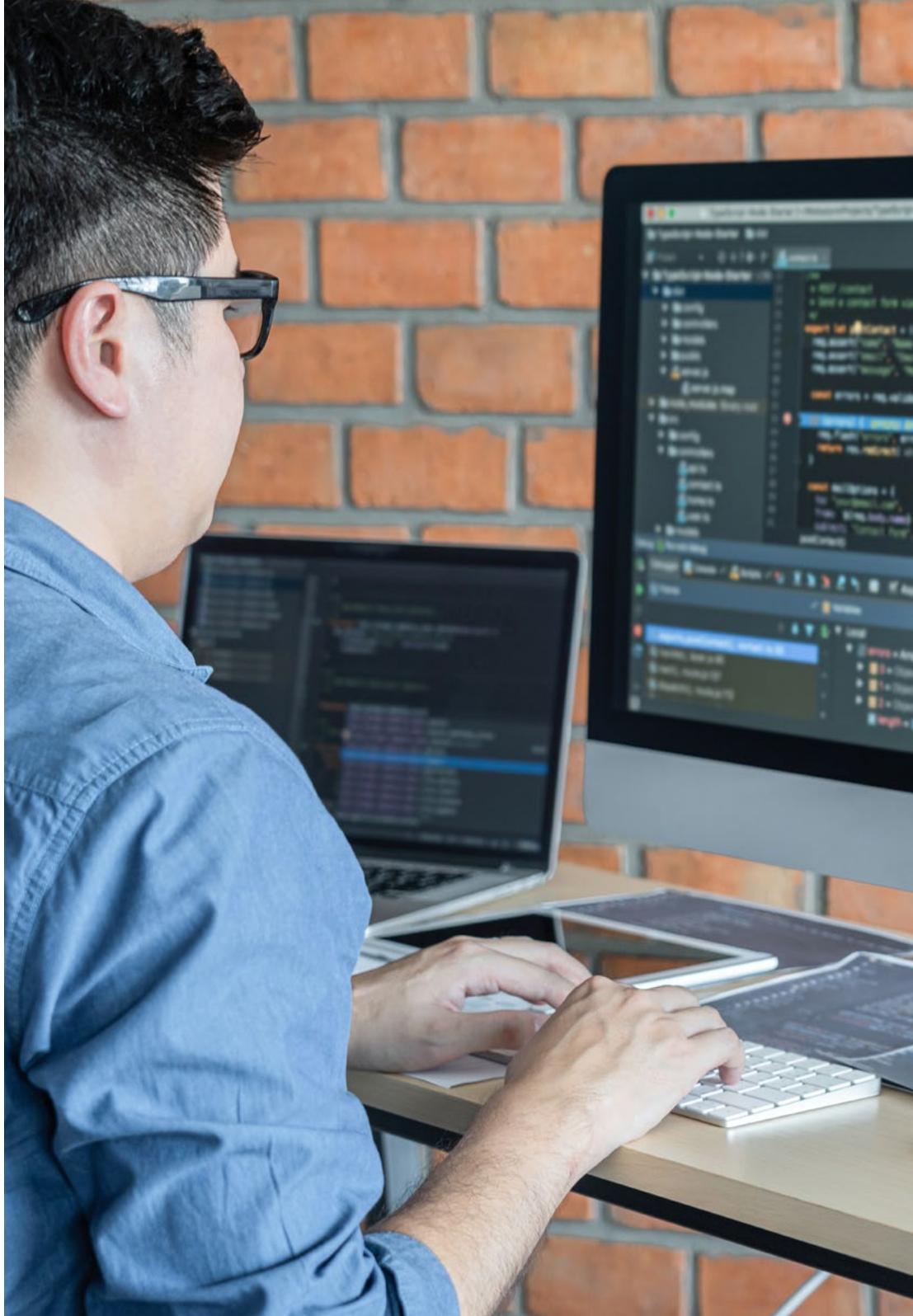
“

Le stage vous offrira l'opportunité de collaborer à des projets réels, tant individuellement qu'en équipe, sous la supervision et l'orientation de professionnels expérimentés du secteur”

La Formation Pratique de ce programme en Programmation Front End Full Stack Developer consiste en un séjour de 3 semaines dans une prestigieuse entreprise informatique, du lundi au vendredi, avec 8 heures consécutives de formation pratique, aux côtés d'un assistant spécialiste. Ce stage lui permettra de travailler sur des projets de programmation réels, aux côtés d'une équipe de professionnels de premier plan dans le domaine des soins infirmiers intensifs, en appliquant les procédures de diagnostic les plus innovantes et en planifiant la dernière génération de thérapie pour chaque pathologie.

Dans cette formation, de nature totalement pratique, les activités visent à développer et à perfectionner les compétences nécessaires pour la Programmation Front End Full Stack Developer et les conditions qui requièrent un haut niveau de qualification, et qui sont orientées vers une formation spécifique pour l'exercice de l'activité, dans un environnement de sécurité pour le patient et de haute performance professionnelle. Il s'agit, sans aucun doute, d'une opportunité d'apprendre en travaillant.

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



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

Module	Activité pratique
Conception et développement d'Interfaces	Créer des interfaces utilisateur intuitives et attrayantes
	Mettre en œuvre une conception réactive qui s'adapte aux différents appareils et tailles d'écran
	Utiliser des outils de conception et de prototypage tels qu'Adobe XD, Figma ou Sketch
	Effectuer des tests de convivialité pour garantir une expérience utilisateur optimale
Développement d'applications Web	Développer des applications web à l'aide de technologies <i>front-end</i> comme HTML, CSS et JavaScript
	Utiliser <i>des frameworks</i> et des bibliothèques comme React, Angular ou Vue.js
	Intégrer des API et des services externes pour enrichir les fonctionnalités des applications
	Optimiser les performances des applications web pour une meilleure expérience utilisateur
Gestion de projets et travail en équipe	Utiliser des méthodologies agiles telles que Scrum ou Kanban pour gérer les projets
	Collaborer avec les concepteurs, les développeurs et les autres membres de l'équipe
	Participer aux revues de code et fournir un <i>feedback</i> constructif
	Gérer le contrôle des versions à l'aide d'outils tels que Git et GitHub
Développement back-end et bases de données	Développer et maintenir des serveurs et des bases de données en utilisant Node.js et Express
	Concevoir et gérer des bases de données relationnelles et non relationnelles telles que MySQL, PostgreSQL et MongoDB
	Mettre en œuvre et sécuriser des API RESTful pour la communication entre le <i>front-end</i> et le <i>back-end</i>
	Gérer l'authentification et l'autorisation des utilisateurs dans les applications web

Module	Activité pratique
Test et débogage	Écrire et exécuter des tests unitaires et d'intégration en utilisant des outils tels que Jest, Mocha ou Jasmine
	Identifier et corriger efficacement les bogues dans le code
	Automatiser les tests et les déploiements à l'aide de pipelines CI/CD
Sécurité et optimisation	Effectuer des revues de code pour maintenir les normes de qualité et les meilleures pratiques
	Mise en œuvre de mesures de sécurité pour protéger les applications web contre les menaces et les vulnérabilités
	Optimiser les performances des applications web, en réduisant les temps de chargement et en améliorant l'efficacité du code
	Mettre en œuvre des techniques d'optimisation SEO pour améliorer la visibilité sur les moteurs de recherche
	Réaliser des audits de sécurité et de performance des applications web



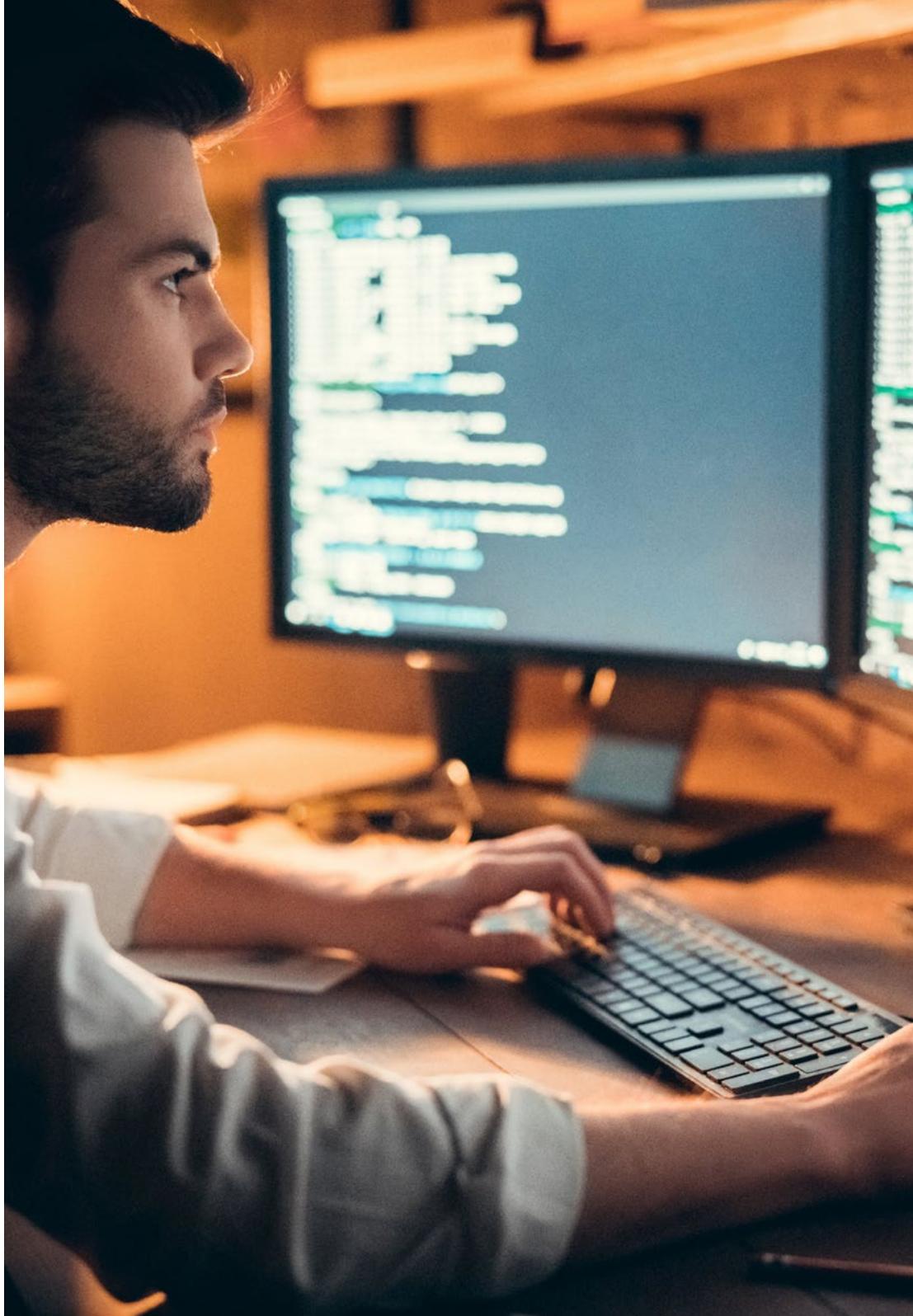
Vous serez confrontés à des défis spécifiques en matière de développement web, de la mise en œuvre de mises en page interactives et de fonctionnalités à l'intégration de bases de données et l'optimisation des performances”

Assurance responsabilité civile

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

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

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



Conditions générales de la formation pratique

Les conditions générales de la convention de stage pour le programme sont les suivantes:

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

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

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

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

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

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

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

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

08

Où puis-je effectuer mon Stage Pratique?

Ce programme de Mastère Hybride inclut dans son itinéraire un séjour pratique dans une prestigieuse entreprise informatique où les étudiants mettront en pratique tout ce qu'ils ont appris dans le domaine de la Programmation Front End Full Stack Developer. Dans ce sens, et afin d'apporter ce diplôme à un plus grand nombre de professionnels, TECH offre au diplômé la possibilité de l'étudier dans différentes institutions. De cette manière, TECH renforce son engagement en faveur d'une éducation de qualité et abordable pour tous.





“

En plus de renforcer vos compétences techniques, les stages favoriseront le développement de vos compétences en matière de communication, de gestion de projet et de résolution de problèmes”

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



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



Informatique

Captia Ingeniería

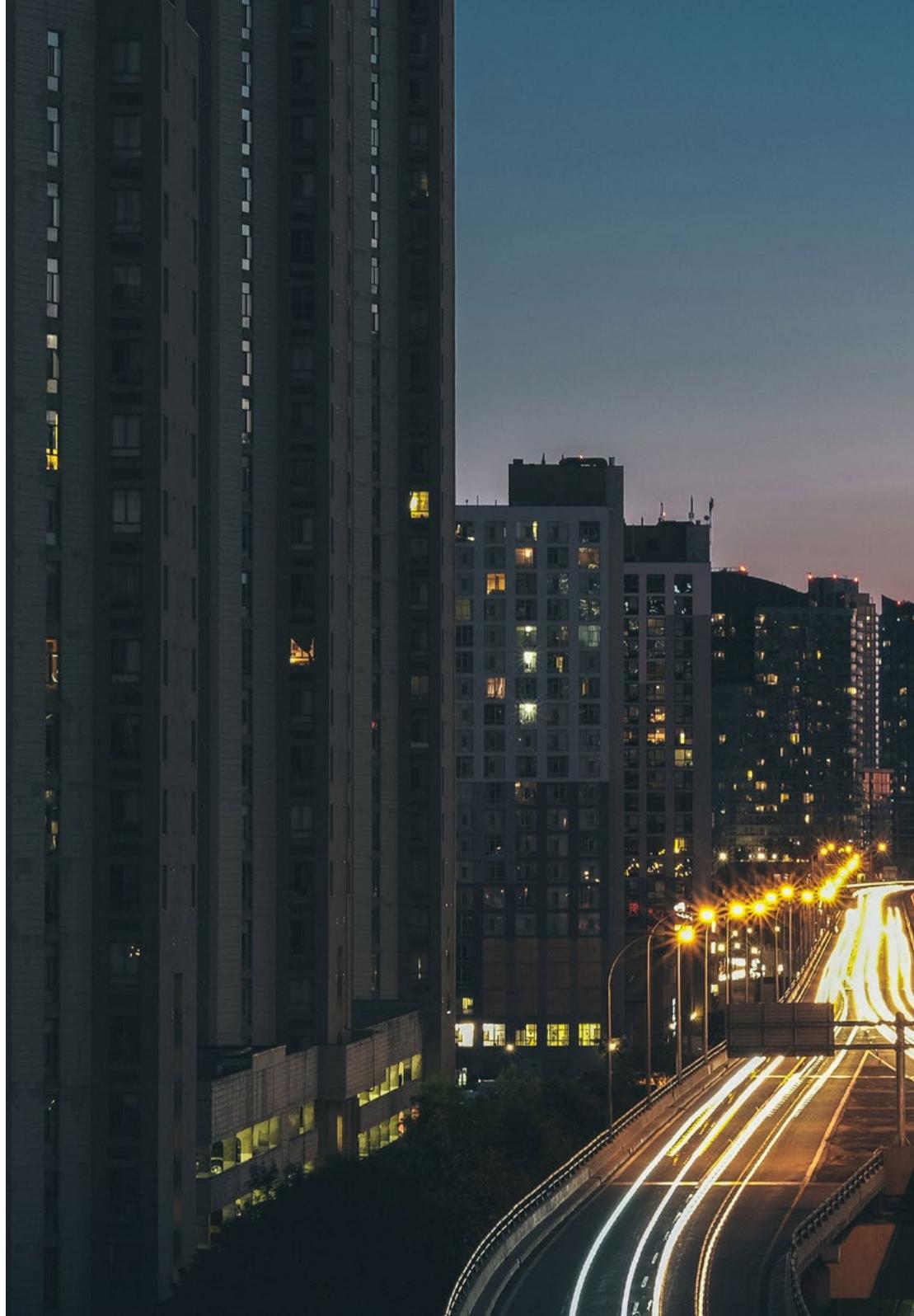
Pays	Ville
Espagne	Madrid

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

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

Formations pratiques connexes:

- Visual Analytics et Big Data
- Développement de Software





“

Boostez votre carrière professionnelle grâce à un enseignement holistique, qui vous permet de progresser à la fois sur le plan théorique et pratique”

09

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.



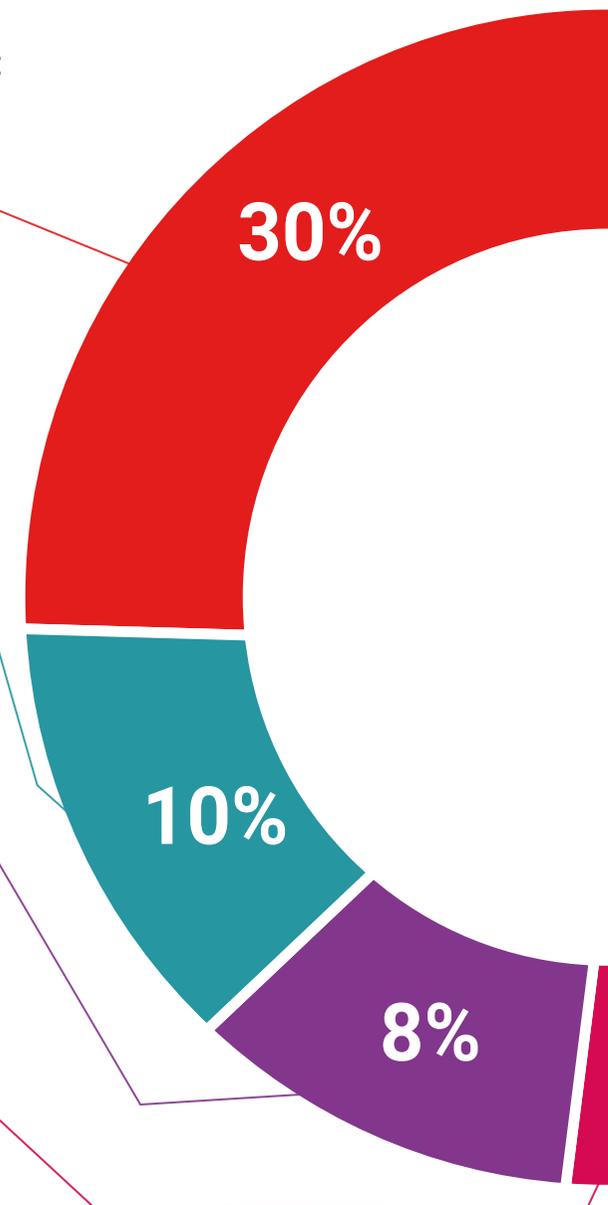
Pratiques en compétences et aptitudes

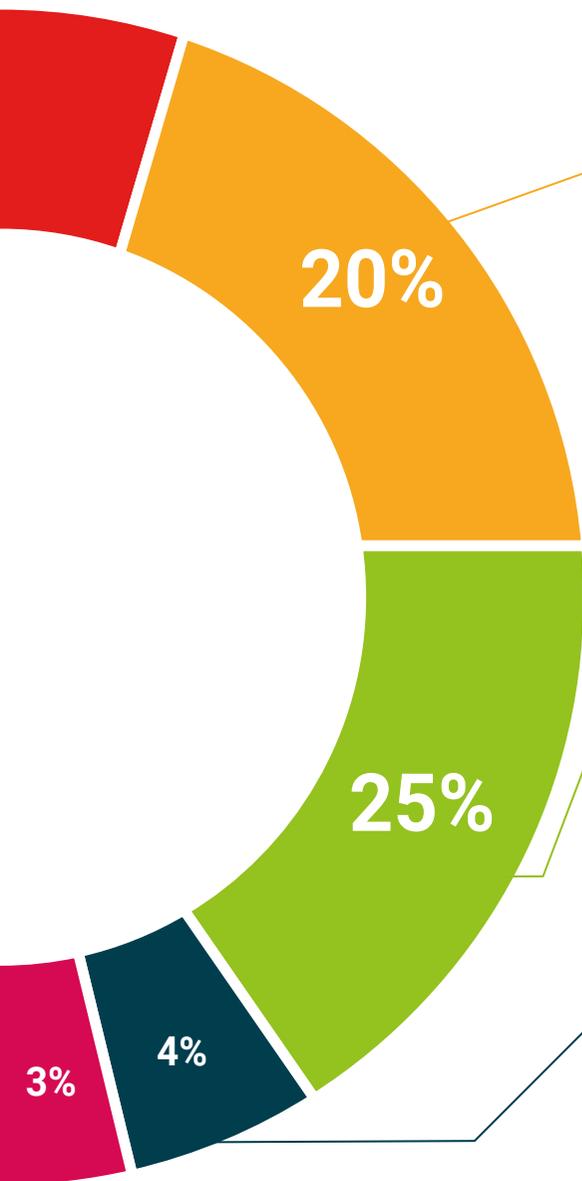
Les étudiants réaliseront des activités visant à développer des compétences et des aptitudes spécifiques dans chaque domaine. Des activités pratiques et dynamiques pour acquérir et développer les compétences et aptitudes qu'un spécialiste doit développer dans le cadre de la mondialisation dans laquelle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



10 Diplôme

Le Diplôme de Mastère Hybride en Programmation Front End Full Stack Developer garantit, outre la formation la plus rigoureuse et la plus actualisée, l'accès à un diplôme de Mastère Hybride délivré par TECH Université Technologique.



“

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

Ce diplôme de **Mastère Hybride en Programmation Front End Full Stack Developer** contient le programme le plus complet et le plus actuel sur la scène professionnelle et académique.

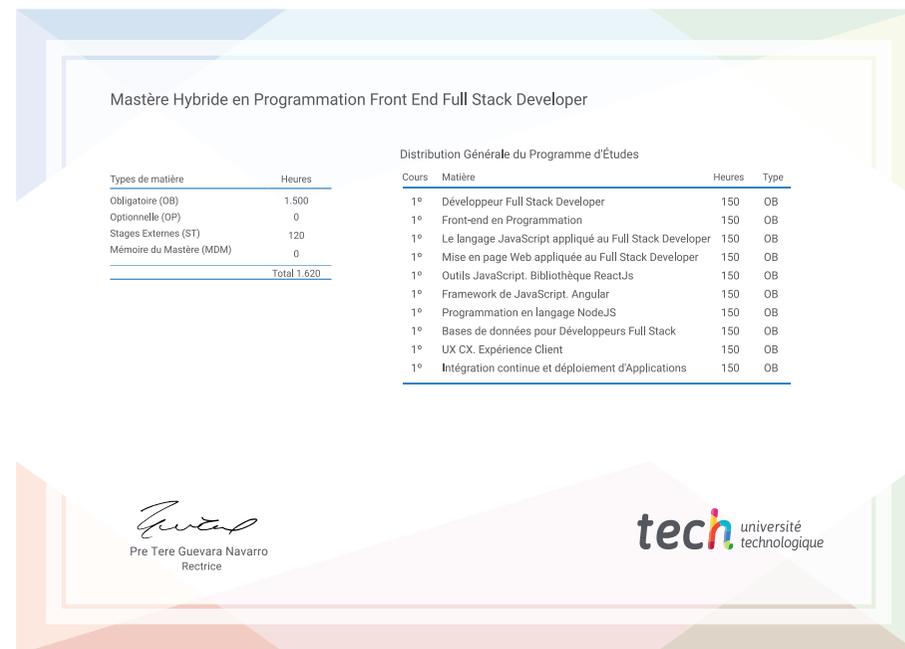
Une fois que l'étudiant aura réussi les évaluations, il recevra par courrier, avec accusé de réception, le diplôme de Mastère Hybride correspondant délivré par TECH.

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

Diplôme: **Mastère Hybride en Programmation Front End Full Stack Developer**

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

Durée: **12 mois**



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

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



Mastère Hybride

Programmation Front End
Full Stack Developer

Modalité: Hybride (en ligne + Stage Pratique)

Durée: 12 mois

Diplôme: TECH Université Technologique

Mastère Hybride

Programmation Front End
Full Stack Developer

