

# Mestrado Próprio

Programação Front End

Full Stack Developer



```
id = '$day->id' AND enabled='y' GROU  
count, "title" => $day_info->title);  
('error studio');  
= '$date') != 1) die('date not found');  
mysql::query("SELECT image.id as image_id FROM image, image_date WHERE image_date.id=image.day id AND image_date.shot_date='$date' AND  
= mysql::fetch($result)) {  
$image->copyright = metadata::get_copyright($image->image_id);  
$image->models = metadata::get_models($image->image_id);  
$return[$image->image_id] = $image;  
}  
1273
```



## Mestrado Próprio

### Programação Front End Full Stack Developer

- » Modalidade: online
- » Duração: 12 meses
- » Certificação: TECH Universidade Tecnológica
- » Créditos: 60 ECTS
- » Tempo Dedicado: 16 horas/semana
- » Horário: ao seu próprio ritmo
- » Exames: online

Acesso ao site: [www.techtute.com/pt/informatica/mestrado-proprio/mestrado-proprio-programacao-front-end-full-stack-developer](http://www.techtute.com/pt/informatica/mestrado-proprio/mestrado-proprio-programacao-front-end-full-stack-developer)

# Índice

01

Apresentação

---

*pág. 4*

02

Objetivos

---

*pág. 8*

03

Competências

---

*pág. 14*

04

Direção do curso

---

*pág. 18*

05

Estrutura e conteúdo

---

*pág. 22*

06

Metodologia

---

*pág. 34*

07

Certificação

---

*pág. 42*

# 01

# Apresentação

Os programadores *Front End* são uma parte indispensável de qualquer desenvolvimento web. A sua natureza de ligação entre a parte de design gráfico e a própria programação *Back End* obriga-os a ter competências altamente específicas e desenvolvidas, as quais, por sua vez, devem ser mantidas atualizadas a fim de responder aos desafios cada vez mais exigentes dos dias de hoje. Sendo uma especialidade que oferece múltiplas oportunidades a nível profissional, a TECH reuniu a melhor equipa docente possível para desenvolver um programa universitário extensivo, completo e metodológico. Apostando nas ferramentas e metodologias mais avançadas no desenvolvimento de soluções web, esta certificação oferece o impulso necessário para dar o passo em direção à programação *Front End Full Stack*.



“

*Torne-se uma parte essencial de qualquer desenvolvimento web, liderando e coordenando equipas informáticas graças à aperfeiçoada metodologia de trabalho e liderança deste Mestrado Próprio”*

O desenvolvimento *Full Stack* é uma opção particularmente interessante para todos os profissionais de informática que procuram impulsionar significativamente as suas carreiras. As competências necessárias para um bom desempenho no setor são extensas, o que significa que as oportunidades de prosperar e até de liderar equipas de programação são múltiplas.

Graças ao rigor com que todos os conteúdos deste programa foram desenvolvidos, os formandos serão capazes de dirigir a sua carreira para o desenvolvimento do *Front End* web, layout da página, especialista em experiência de clientes ou DevOps. Com uma visão de 360° de todo o processo de construção de uma aplicação/web, o informático será capaz de enfrentar qualquer tipo de projeto, proporcionando também um desenvolvimento nos últimos avanços em todos os processos do ciclo de vida de um software.

Uma oportunidade académica única de acesso ao conhecimento que combina a mais recente teoria informática com a prática profissional de alto nível, proporcionada por uma equipa docente da mais alta qualidade. A sua experiência à frente de numerosos projetos relevantes no campo da banca digital ou das telecomunicações torna o conteúdo didático mais rico, fornecendo um grande número de casos reais e leituras complementares.

A grande flexibilidade deste ensino é outra das suas características mais marcantes. Não há horários fixos ou aulas presenciais, e o aluno decide quando, onde e como assumir a carga horária completa do curso. Todo o conteúdo da aula virtual está disponível para download e pode ser estudado a partir de qualquer dispositivo com uma ligação à Internet.

Este **Mestrado Próprio em Front End Full Stack Developer** conta com o conteúdo educacional mais completo e atualizado do mercado. As suas principais características são:

- ◆ O desenvolvimento de estudos de caso apresentados por especialistas em front end full stack developer programming.
- ◆ O conteúdo gráfico, esquemático e eminentemente prático do livro fornece informações científicas e práticas sobre as disciplinas que são essenciais para a prática profissional
- ◆ Exercícios práticos onde o processo de autoavaliação pode ser levado a cabo a fim de melhorar a aprendizagem
- ◆ A sua ênfase especial em metodologias inovadoras
- ◆ Lições teóricas, questões ao especialista e trabalhos de reflexão individual
- ◆ A disponibilidade de acesso ao conteúdo a partir de qualquer dispositivo fixo ou portátil com ligação à internet



*Inscreva-se agora e não perca a oportunidade que o levará ao auge na liderança e desenvolvimento dos projetos informáticos mais ambiciosos”*

“

*Alcançará um nível avançado de especialização, sendo capaz de construir qualquer solução web necessária com uma perspetiva de Experiência do Cliente moderna e adaptada ao mercado atual”*

O corpo docente do curso inclui profissionais do setor que trazem a sua experiência profissional para este curso, para além de especialistas reconhecidos de sociedades de referência e universidades de prestígio.

Graças ao seu conteúdo multimédia, desenvolvido com a mais recente tecnologia educacional, o profissional terá acesso a uma aprendizagem situada e contextual, ou seja, um ambiente de simulação que proporcionará um programa imersivo programado para se formar em situações reais.

A conceção deste programa baseia-se na Aprendizagem Baseada nos Problemas, através da qual o instrutor deve tentar resolver as diferentes situações da atividade profissional que surgem ao longo do Programa académico. Para tal, contará com a ajuda de um sistema inovador de vídeo interativo desenvolvido por especialistas reconhecidos.

*Aprofundará a sua compreensão das metodologias ágeis e como podem ser implementadas no processo de desenvolvimento, aumentando as suas aptidões e competências transversais.*

*Terá acesso a uma vasta gama de material didático, desde a linguagem de programação Javascript até ferramentas como CSS, Angular e ReactJS.*



# 02

## Objetivos

O objetivo deste Mestrado Próprio em Programação Front End Full Stack Developer, tendo em conta as múltiplas opções que o desenvolvimento oferece, não é outro senão o de fornecer os conhecimentos e técnicas mais avançadas neste campo. Desta forma, os informáticos podem mesmo começar a desenvolver os seus próprios projetos ou impulsionar as suas carreiras profissionais antes de completarem o curso, graças à abordagem eminentemente prática de todos os conteúdos fornecidos.





“

*Atingirá os seus objetivos de carreira mais ambiciosos graças à abordagem distinta deste programa, que o levará através de todas as fases do desenvolvimento Front End Full Stack”*



## Objetivos gerais

---

- ◆ Gerar conhecimentos especializados sobre questões-chave de programação
- ◆ Fomentar o pensamento de forma algorítmica
- ◆ Fornecer as ferramentas e competências necessárias para o desenvolvimento
- ◆ Promover a adoção de metodologias ágeis para a execução de projeto
- ◆ Desenvolver conhecimentos especializados sobre os fundamentos da web
- ◆ Promover a utilização de técnicas e ferramentas de desenvolvimento *Front End* modernas
- ◆ Desenvolver um design web para tratar corretamente o layout
- ◆ Avaliar os conhecimentos adquiridos





## Objetivos específicos

---

### Módulo 1. Desenvolvimento *Full Stack Developer*

- ◆ Desenvolver capacidades avançadas de programação
- ◆ Incentivar a utilização de sistemas de controlo de versões e plataformas de alojamento de código
- ◆ Promover a utilização de Metodologias Ágeis
- ◆ Aprofundar os conceitos-chave e o funcionamento da Internet
- ◆ Aumentar a destreza na linha de comandos

### Módulo 2. *Front-end* em Programação

- ◆ Identificar e compreender a sintaxe correta de HTML e CSS
- ◆ Explorar os diferentes elementos do HTML
- ◆ Determinar a abordagem de desenho adaptativo
- ◆ Utilizar formato de apresentação para páginas web, aplicando folhas de estilo em cascata
- ◆ Incorporar pré-processador de CSS
- ◆ Estabelecer os benefícios da utilização de um pré-processador
- ◆ Gerar conhecimento especializado sobre sistemas de desenho
- ◆ Estabelecer critérios para a utilização do sistema de desenho

### Módulo 3. Linguagem JAVASCRIPT aplicada a *Full Stack Developer*

- ◆ Estabelecer os tipos básicos e complexos oferecidos pelo JavaScript
- ◆ Analisar as diferentes formas de programação com a linguagem e fazer o uso correto em cada situação.
- ◆ Atualizar os conhecimentos para as versões mais recentes
- ◆ Descobrir a programação funcional
- ◆ Examinar a programação assíncrona e as suas características

### Módulo 4. Web layout aplicada a *Full Stack Developer*

- ◆ Avaliar um desenho web a fim de saber como situá-lo temporariamente
- ◆ Examinar as principais regras CSS
- ◆ Apresentar diferentes metodologias CSS para obter desenhos *Responsive*
- ◆ Fundamentar os princípios de desenvolvimento em cascata CSS
- ◆ Identificar a tecnologia Bootstrap em qualquer desenho web
- ◆ Analisar os princípios de Bootstrap
- ◆ Desenvolver um layout web usando Bootstrap
- ◆ Determinar os princípios de desenvolvimento de um projeto SaSS

### Módulo 5. Ferramentas de JavaScript. Biblioteca ReactJs

- ◆ Determinar as funcionalidades de React
- ◆ Configurar um projeto utilizando o Create React App
- ◆ Analisar o ciclo de vida dos componentes em React
- ◆ Gerar conhecimentos especializados sobre as modernas funcionalidades de React, tais como *Hooks* e *Context*
- ◆ Estabelecer estados globais usando *Context*
- ◆ Criar e renderizar listas e criar formulários com React
- ◆ Implementar validação de campo nos formulários
- ◆ Estilizar componentes e elementos
- ◆ Depurar, testar e implementar aplicações React

### Módulo 6. *Framework* de JavaScript. Angular

- ◆ Desenvolver conhecimentos especializados sobre a arquitetura *Framework*
- ◆ Aprofundar a metodologia de Angular
- ◆ Analisar o conceito de componentes
- ◆ Organizar o código corretamente

### Módulo 7 Programação em linguagem NodeJS

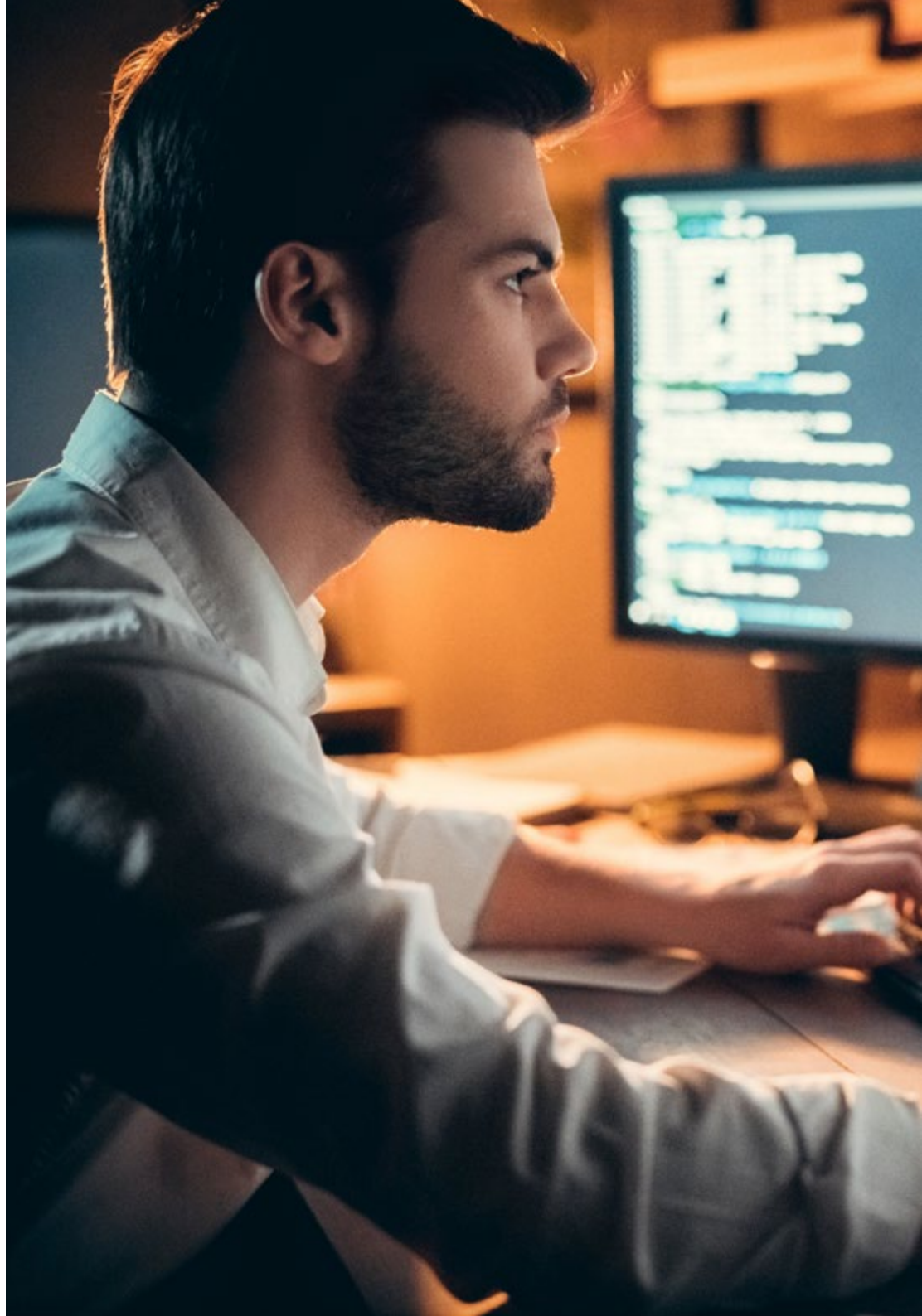
- ◆ Gerar conhecimentos especializados sobre os tipos de JavaScript e os seus operadores
- ◆ Analisar as melhores formas de programar com a linguagem
- ◆ Atualizar os conhecimentos para as versões mais recentes
- ◆ Explorar a programação funcional
- ◆ Desenvolver a programação assíncrona e a sua motivação
- ◆ Adquirir a capacidade de realizar uma aplicação com NodeJS Index

### Módulo 8. Bases de dados para programadores *Full Stack*

- ◆ Determinar porque utilizar uma base de dados no desenvolvimento de aplicações
- ◆ Examinar os tipos de bases de dados disponíveis e as suas diferenças
- ◆ Desenvolver uma ideia clara do que cada tipo de base de dados é utilizado para
- ◆ Analisar a utilização de bases de dados nos atuais paradigmas de desenvolvimento

### Módulo 9. UX CX. Experiência do cliente

- ◆ Analisar a importância do utilizador hoje em dia e aprofundar a cultura de feedback
- ◆ Realizar estratégias de omni-canal e personalização baseadas em microinterações
- ◆ Estudar a evolução da análise web até à análise comportamental
- ◆ Determinar como a Inteligência Artificial levou o CX ao nível seguinte



- ◆ Estabelecer as técnicas mais importantes de análise de experiência web, mobilidade e acessibilidade
- ◆ Apresentar a metodologia *Design Thinking* e o processo de criação de experiência do utilizador
- ◆ Desenvolver ferramentas concretas de prototipagem e *WireFraming*, assim como *Frameworks* de desenvolvimento de *Front-End*

### Módulo 10. Integração contínua e implantação de aplicações

- ◆ Concretizar os benefícios da adoção de um modelo automático de implementação de aplicações
- ◆ Estabelecer as diferenças entre integração contínua, entrega e a implementação contínuas
- ◆ Determinar as principais características de DevOps
- ◆ Avaliar algumas das ferramentas-chave para implementar pipelines de CI/CD
- ◆ Desenvolver os fatores essenciais para o desenvolvimento de aplicações prontas a apoiar os processos CI/CD
- ◆ Examinar as tecnologias de contentores como um pilar fundamental da prática da CI/CD

“Melhorará as suas aptidões e competências de forma progressiva, ao longo de 10 módulos elaborados a partir dos conhecimentos mais sólidos e da experiência comprovada de todos os docentes”

# 03

## Competências

As competências que deve desenvolver um programador *Front End Full Stack* são múltiplas, especialmente num campo tão disputado e especializado como o da informática. Por esta razão, o programa abrange os diferentes tipos de linguagem mais comuns neste campo, bem como as ferramentas e filosofia de trabalho que o informático deve seguir para se destacar com distinção profissional. Isto é possível graças à natureza multidisciplinar da própria equipa docente, que reuniu todos os seus conhecimentos em várias áreas de desenvolvimento *Front End*.



“

*Dará ao seu CV um impulso de alta-qualidade ao incorporar uma multiplicidade de aptidões e competências necessárias nos escalões superiores de desenvolvimento de software”*



## Competências gerais

- ◆ Reconhecer corretamente a sintaxe das linguagens HTML e CSS
- ◆ Desenvolver critérios de boas práticas para a programação web
- ◆ Gerar conhecimentos especializados sobre a linguagem JavaScript
- ◆ Ser capaz de desenvolver qualquer tipo de aplicação com JavaScript
- ◆ Analisar a biblioteca Bootstrap
- ◆ Levar a cabo projetos de layout com SaSS (*Syntactically Awesome Stylesheets*)
- ◆ Identificar a sintaxe de React e como programar usando-a
- ◆ Aplicar boas práticas à linguagem
- ◆ Examinar o processo de carga e acesso em cada um dos principais tipos de bases de dados da sua área
- ◆ Avaliar as ferramentas e técnicas mais importantes na análise de CX e o 'stack tecnológico' habitual nas empresas







## Competências específicas

---

- ♦ Analisar as diferentes estruturas de dados
- ♦ Examinar as técnicas de desenho e interpretação de algoritmos
- ♦ Preparar o ambiente de desenvolvimento
- ♦ Clonar um website
- ♦ Construir um website com Bootstrap
- ♦ Compilar código CSS com SaSS
- ♦ Desenvolver um *Framework* CSS próprio baseado em Bootstrap usando SaSS
- ♦ Gerar um projeto e pô-lo em ação
- ♦ Estabelecer como ligar e carregar/extrair dados de diferentes tipos de bases de dados
- ♦ Identificar práticas, casos de utilização, tecnologias e ferramentas do ecossistema CI/CD, essenciais para apoiar todo o processo no seu conjunto



*Desenvolverá ao longo do curso as competências necessárias para ter sucesso no desenvolvimento Front End, reforçando não só os seus próprios conhecimentos mas também as suas competências transversais”*



A hand is shown typing on a laptop keyboard in the foreground. In the background, a computer monitor displays lines of code, and a desk with a calculator and other office supplies is visible. The image is partially obscured by a teal and white geometric overlay.

“

*Poderá consultar todas as suas dúvidas diretamente com o pessoal docente, obtendo uma tutoria personalizada e adaptada às suas necessidades”*

## Direção



### Sr. Martín Olalla Bonal

- ◆ Especialista Técnico em *Blockchain* na IBM SPGI
- ◆ Especialista em vendas técnicas em *Blockchain*. IBM
- ◆ Diretor de Arquitetura *Blocknitive*
- ◆ Técnico em Eletrónica Digital
- ◆ *Blockchain Architect* -*IT Infrastructure Architect* - Gestor de projetos IT. Áreas de negócio: Software, Infraestrutura, Telecomunicações

## Professores

### Sr. Arturo Guerrero Díaz-Pintado,

- ◆ Engenheiro Técnico de Pré-Venda através da *Watson Customer Engagement portfolio* (soluções de *Marketing* e *Customer Experience*) em Espanha, Portugal, Grécia, Israel na IBM
- ◆ Engenheiro de redes de I+D na Telefónica
- ◆ Consultor de serviços profissionais que trabalha com organizações de vanguarda na Europa, Médio Oriente e América Latina desde a IBM
- ◆ Licenciado em Engenharia Superior de Telecomunicações pela Universidade de Alcalá e pela *Danish Technical University*
- ◆ Colaborações em universidades de renome e centros de ensino superior em temas relacionados com a tecnologia, tais como *Inteligência Artificial*, *Internet of Things*, *Cloud*, *Customer Experience* e *Digital Transformation*

### Sr. Jesús Calzada Martínez,

- ◆ Engenheiro de Software Sênior na Devo
- ◆ Programador *Full Stack* na *Blocknitive*
- ◆ Responsável de *Front End* na *Infinia*
- ◆ Programador *Full Stack* na *Resem*
- ◆ Programador Java na *Hitec*
- ◆ Licenciado em Engenharia de Computadores

**Sr. Bruno Pintado San Claudio**

- ◆ Coordenador de Desenvolvimento em IDavinci
- ◆ Java Developer na Biblioteca Nacional de Espanha
- ◆ Programador de suporte e técnico de redes N1 na Sanitas
- ◆ Técnico de apoio de sistemas no Município de Alcobendas
- ◆ Técnico de Comunicações N1 para ADIF no Centro de Telecomunicações de Atocha
- ◆ Certificado em Engenharia Técnica de Telecomunicações com especialidade em Sistemas Eletrónicos na Universidade Politécnica de Madrid
- ◆ Certificado em Engenharia Eletrónica de Comunicações pela Universidade Politécnica de Madrid

**Sr. Luis Reyes Oliva**

- ◆ Programador de desenvolvimento e arquiteto cloud na IBM
- ◆ Responsável técnico de clientes para contas integradas do BBVA na IBM
- ◆ *Cloud Executive Selling* na IBM
- ◆ Arquiteto Cloud e DevOps na IBM
- ◆ Arquiteto de software para clientes na Telefónica
- ◆ Arquiteto de soluções técnicas para a Rational
- ◆ *Software Engineering Manager* na Borland
- ◆ Desenvolvimento Software e responsável de controlo de qualidade na Altana Consulting
- ◆ Licenciado em Engenharia Informática na Universidade Pontifícia de Salamanca, em Madrid

**Sr. Pedro Luis Frias Favero**

- ◆ CTO - Swearit Technologies
- ◆ COO - Key Identification
- ◆ Especialista em *blockchain* e aplicações descentralizadas - Universidade de Alcalá
- ◆ Full Stack Developer - Ironhack
- ◆ Engenheiro Industrial licenciado pela Universidade de Yacambu

**Sr. Antonio Gómez Rodríguez**

- ◆ Engenheiro de Soluções Cloud na Oracle
- ◆ Diretor de Projetos em Sopra Group
- ◆ Diretor de Projetos em Everis
- ◆ Chefe de Projetos na Empresa Pública de Gestão de Programas Culturais Secretaria de Cultura de Andaluzia Secretaria de Cultura de Andaluzia
- ◆ Analista de Sistemas de Informação. Sopra Group
- ◆ Licenciado em Engenharia de Telecomunicações pela Universidade Politécnica da Catalunha
- ◆ Pós-graduação em Tecnologias e Sistemas de Informação do Instituto Catalão de Tecnologia
- ◆ *E-Business Master* pela Escola de Negócios La Salle

# 05

## Estrutura e conteúdo

Dado que o esforço e o investimento feitos pelo informático ao optar por um programa desta natureza, a TECH está empenhada numa metodologia de ensino de primeira classe, comprovada em termos de eficiência e rentabilidade. O *Relearning* permite ao estudante adquirir os conhecimentos mais importantes do programa ao longo de todo o programa, evitando assim um tempo de estudo valioso que pode ser gasto nos muitos materiais suplementares que acompanham cada tópico proposto.



“

*Explore os diferentes módulos e tópicos de conhecimento através de resumos em vídeo, vídeos detalhados e vídeos motivacionais criados pelos próprios docentes”*

## Módulo 1. Desenvolvimento *Full Stack Developer*

- 1.1. Desenvolvimento *Full Stack Developer* I. Programação e linguagens
  - 1.1.1. Programação
  - 1.1.2. Funções de programação
  - 1.1.3. Linguagens e *Framework*
  - 1.1.4. Algoritmo
  - 1.1.5. Características de um algoritmo
- 1.2. Desenvolvimento *Full Stack Developer* II. Tipologia
  - 1.2.1. Variáveis e constantes
  - 1.2.2. Tipos
  - 1.2.3. Operadores
  - 1.2.4. Declarações
  - 1.2.5. Laços
  - 1.2.6. Funções e objetivos
- 1.3. Estruturas de dados no desenvolvimento
  - 1.3.1. Tipos de estrutura lineares
  - 1.3.2. Tipos de estrutura funcionais
  - 1.3.3. Tipos de estrutura de árvore
- 1.4. Desenho e interpretação de algoritmos
  - 1.4.1. Paralelismo no desenvolvimento. Divida e vencerá
  - 1.4.2. Algoritmos vorazes
  - 1.4.3. Programação dinâmica
- 1.5. Ambiente e instrumentos para o desenvolvimento orientado ao *Full Stack Developer*
  - 1.5.1. Preparação do ambiente para Mac OS
  - 1.5.2. Preparação do ambiente para Linux
  - 1.5.3. Preparação do ambiente para Windows
- 1.6. Linha de comando. Tipologia e funcionamento
  - 1.6.1. O terminal
  - 1.6.2. Emuladores
  - 1.6.3. Intérprete de comandos
  - 1.6.4. Primeiros comandos
  - 1.6.5. Navegação
  - 1.6.6. Gestão de ficheiros e pastas utilizando a interface de linha de comandos
  - 1.6.7. *Secure Shell*. SSH
  - 1.6.8. Comandos avançados
- 1.7. Git. Repositório de software
  - 1.7.1. Git Repositório de software
  - 1.7.2. Uso de Git
  - 1.7.3. Repositório de software
  - 1.7.4. Ramos
  - 1.7.5. Ciclo de trabalho
  - 1.7.6. Comandos
- 1.8. Serviço de alojamento de versões de código
  - 1.8.1. Serviço de alojamento de versões de código
  - 1.8.2. Fornecedores
  - 1.8.3. Repositórios
- 1.9. Internet
  - 1.9.1. Internet
  - 1.9.2. Protocolos utilizados em WWW
  - 1.9.3. Protocolo HTTP
- 1.10. Metodologias em desenvolvimento *Full Stack*
  - 1.10.1. *Scrum*
  - 1.10.2. XP
  - 1.10.3. *Design Sprint*



## Módulo 2. *Front End* em Programação

- 2.1. Linguagem HTML
  - 2.1.1. Documento HTML
  - 2.1.2. Elemento *Head*
  - 2.1.3. Elemento *Body*
  - 2.1.4. Texto
  - 2.1.5. Hipervínculos
  - 2.1.6. Imagens
  - 2.1.7. *Primer Site*
- 2.2. Linguagem HTML. *Layouts*
  - 2.2.1. Linguagem HTML. Elementos
  - 2.2.2. *Layout* tradicional
  - 2.2.3. *Layout* semântico
- 2.3. Folhas de Estilo em Cascata CSS (*Cascading Style Sheets*)
  - 2.3.1. Inclusão de CSS num documento HTML
  - 2.3.2. Comentários
  - 2.3.3. Seletores
  - 2.3.4. Seletores avançados
- 2.4. Propriedades CSS (*Cascading Style Sheets*)
  - 2.4.1. Cor
  - 2.4.2. Texto
  - 2.4.3. Pseudo-classes
  - 2.4.4. Transições
  - 2.4.5. Animações
  - 2.4.6. Animação de elementos
  - 2.4.7. Animação avançada
- 2.5. Modelo de caixa
  - 2.5.1. Altura e largura
  - 2.5.2. Margem
  - 2.5.3. Preenchimento
- 2.6. Posicionamento
  - 2.6.1. Posicionamento estático
  - 2.6.2. Posicionamento relativo
  - 2.6.3. Posicionamento absoluto
  - 2.6.4. Posicionamento fixo
  - 2.6.5. *Floats*
- 2.7. Desenho adaptável
  - 2.7.1. *Viewport*
  - 2.7.2. *Media Queries*
  - 2.7.3. Unidades CSS
  - 2.7.4. *Imagens*
  - 2.7.5. *Frameworks*
- 2.8. *Layout* moderno
  - 2.8.1. *Flex*
  - 2.8.2. *Grid*
  - 2.8.3. *Flex vs. Grid*
- 2.9. Pré-processador
  - 2.9.1. Sass
  - 2.9.2. Variáveis
  - 2.9.3. Mixins
  - 2.9.4. Laços
  - 2.9.5. Funções
- 2.10. Sistema de desenho
  - 2.10.1. Bootstrap
  - 2.10.2. Grelha de Bootstrap
  - 2.10.3. *Header* e *Footer* do nosso site
  - 2.10.4. Formulários
  - 2.10.5. *Cards*
  - 2.10.6. Maneiras

### Módulo 3. Linguagem JAVASCRIPT aplicada a *Full Stack Developer*

- 3.1. Tipos primitivos e operadores
  - 3.1.1. Linguagem JavaScript
  - 3.1.2. Números e os seus operadores
  - 3.1.3. Cadeias de texto e os seus operadores
  - 3.1.4. Valores booleanos
  - 3.1.5. Conversão entre tipos
- 3.2. Controladores de fluxo e estrutura
  - 3.2.1. Expressões e sentenças
  - 3.2.2. Variáveis e constantes
  - 3.2.3. Sentença *If*
  - 3.2.4. Sentenças *For*, *While*
- 3.3. Funções
  - 3.3.1. Funções
  - 3.3.2. Parâmetros
  - 3.3.3. Funções como parâmetros
  - 3.3.4. Âmbitos das variáveis
  - 3.3.5. *Scopes* aninhados
  - 3.3.6. *Hoisting*
  - 3.3.7. *Closures*
  - 3.3.8. Recursividade
- 3.4. Estruturas de dados: objetos
  - 3.4.1. Tipo *Object*
  - 3.4.2. Criação de objetos
  - 3.4.3. Acesso aos valores de um objeto
  - 3.4.4. Adição ou remoção de propriedades
  - 3.4.5. Objetos aninhados
  - 3.4.6. *Destructuring* de objetos
  - 3.4.7. Métodos do tipo *Object*
  - 3.4.8. *Spread Operator*
  - 3.4.9. Imutabilidade
- 3.5. Estrutura de dados: *Array*
  - 3.5.1. Estrutura de dados. *Array*
  - 3.5.2. *Array*. Tipologia
  - 3.5.3. *Arrays* aninhados
  - 3.5.4. Métodos de um *Array*
- 3.6. POO: *Prototype* e classes
  - 3.6.1. POO. Programação orientada a objetos
  - 3.6.2. Protótipos
  - 3.6.3. Classes
  - 3.6.4. Dados privados
  - 3.6.5. Subclasses
  - 3.6.6. *Call* e *Apply*
- 3.7. Tipos de JavaScript
  - 3.7.1. *Set*
  - 3.7.2. *WeakSet*
  - 3.7.3. *Map*
  - 3.7.4. *WeakMap*
  - 3.7.5. Expressões regulares
- 3.8. Utilidades de JavaScript
  - 3.8.1. *Date*
  - 3.8.2. *Math*
  - 3.8.3. *Symbol*
  - 3.8.4. JSON
- 3.9. JavaScript no *Browser*
  - 3.9.1. Inclusão de JavaScript num website
  - 3.9.2. DOM
  - 3.9.3. Eventos
  - 3.9.4. *Storage* no navegador
- 3.10. Programação assíncrona
  - 3.10.1. A programação assíncrona
  - 3.10.2. *Event Loop*
  - 3.10.3. *Callbacks*
  - 3.10.4. *Promises*
  - 3.10.5. *Async/Await*

## Módulo 4. Web layout aplicada a *Full Stack Developer*

- 4.1. CSS e layout
  - 4.1.1. Layout com tabelas
  - 4.1.2. Desenho fluído
  - 4.1.3. A era do *Responsive*
  - 4.1.4. *Mobile First* vs. *Desktop First*
- 4.2. CSS e as regras do desenho web
  - 4.2.1. Seletores
  - 4.2.2. Pseudo-classes
  - 4.2.3. Pseudo-elementos
- 4.3. Layout com CSS
  - 4.3.1. Regras do *Box Model*
  - 4.3.2. Tipografias
  - 4.3.3. Cores
  - 4.3.4. Imagens
  - 4.3.5. Fundos
  - 4.3.6. Tabelas
  - 4.3.7. Formulários
  - 4.3.8. Mostrar e esconder elementos
  - 4.3.9. Variáveis CSS
- 4.4. Desenho *Responsive* e desenho fluído
  - 4.4.1. Elementos flutuantes
  - 4.4.2. *Grid* CSS
  - 4.4.3. *Media Queries*
  - 4.4.4. *Flex Box*
- 4.5. A cascata de CSS
  - 4.5.1. Prioridade de regras CSS
  - 4.5.2. Sobrescrever regras
  - 4.5.3. Classes vs. Identificadores
- 4.6. SaSS
  - 4.6.1. Software como serviço (SaSS)
  - 4.6.2. Instalação de SaSS
  - 4.6.3. Executando e compilando SaSS
  - 4.6.4. Estrutura de um diretório SaSS
- 4.7. Uso de SaSS
  - 4.7.1. Variáveis em Sass
  - 4.7.2. Modularização do nosso projeto
  - 4.7.3. Sintaxe de SaSS
- 4.8. Lógica de SaSS
  - 4.8.1. Mixins
  - 4.8.2. Maps
  - 4.8.3. Funções e estruturas de controlo
- 4.9. Layout com Bootstrap
  - 4.9.1. Bootstrap
  - 4.9.2. *Layout* de Bootstrap
  - 4.9.3. Formulários
  - 4.9.4. *Box Model* com Bootstrap
  - 4.9.5. Cores e tipografias
  - 4.9.6. Ligações e botões
  - 4.9.7. Mostrar e ocultar elementos com Bootstrap
  - 4.9.8. *Flex Box* com Bootstrap
  - 4.9.9. Componentes
- 4.10. *Theming* Bootstrap
  - 4.10.1. Reescrevendo Bootstrap com SaSS (*Software as a Service*)
  - 4.10.2. Estrutura de arquivos
  - 4.10.3. Criação da nossa própria *Framework* CSS (*Cascading Style Sheets*)

## Módulo 5. Ferramentas de JavaScript. Biblioteca ReactJs

- 5.1. Ferramenta de Javascript ReactJS
  - 5.1.1. A ferramenta ReactJS
  - 5.1.2. Create React App
  - 5.1.3. JavaScript *Syntax Extension*
- 5.2. Componentes de ReactJS
  - 5.2.1. Componentes
  - 5.2.2. *Props*
  - 5.2.3. Renderização
- 5.3. Eventos na biblioteca ReactJs
  - 5.3.1. Gestão de eventos
  - 5.3.2. Gestão de eventos em linha
  - 5.3.3. Eventos na biblioteca ReactJs
- 5.4. Configurando *Hooks* de ReactJS
  - 5.4.1. Estados de um componente
  - 5.4.2. *Hook* de estado
  - 5.4.3. *Hook* de efeito
  - 5.4.4. *Custom Hooks*
  - 5.4.5. Outros *Hooks*
- 5.5. *Component Context* em ReactJS
  - 5.5.1. *Component Context* em ReactJS
  - 5.5.2. Uso de *Context*
  - 5.5.3. Estrutura do *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. Aplicação prática de uso de *Context*





- 5.6. Routing em ReactJs
  - 5.6.1. *Router*
  - 5.6.2. *React Router*
  - 5.6.3. Instalação
  - 5.6.4. Routing básico
  - 5.6.5. Routing dinâmico
  - 5.6.6. Componentes primários
  - 5.6.7. *React Router Hooks*
- 5.7. Uso de listas e formulários com ReactJS
  - 5.7.1. Listas e loops
  - 5.7.2. Formulários e validações
  - 5.7.3. *React Hook Forms*
- 5.8. Uso de estilos em ReactJS
  - 5.8.1. Estilo tradicional
  - 5.8.2. Estilos em linha
  - 5.8.3. Adição da biblioteca de sistema de desenho
- 5.9. Realização de testes em Javascript. Ferramentas
  - 5.9.1. *Testing*
  - 5.9.2. *Jest JavaScript Testing Framework*
  - 5.9.3. *Visual Testing and Documentation*
- 5.10. Implementação de código com ReactJS
  - 5.10.1. *Hosting*
  - 5.10.2. Fornecedores
  - 5.10.3. Preparação do projeto
  - 5.10.4. Implementação em Heroku

## Módulo 6. Framework de JavaScript. Angular

- 6.1. O Framework angular e a sua arquitetura
  - 6.1.1. Angular CLI
  - 6.1.2. Arquitetura
  - 6.1.3. *Workspace* e estrutura
  - 6.1.4. Ambiente
- 6.2. Componentes do Framework Angular
  - 6.2.1. Ciclo de vida
  - 6.2.2. Encapsulação da vista
  - 6.2.3. Interação entre componentes
  - 6.2.4. Projeção de conteúdo
- 6.3. Modelos do Framework Angular
  - 6.3.1. Interpolação de texto
  - 6.3.2. Declarações
  - 6.3.3. *Property Binding*
  - 6.3.4. *Class, Style* e *Attribute Binding*
  - 6.3.5. *Event Binding* e *Two-Way Binding*
  - 6.3.6. *Pipes*
- 6.4. Diretivas do Framework Angular
  - 6.4.1. Diretivas de Angular
  - 6.4.2. Diretivas de atributo
  - 6.4.3. Diretivas de estrutura
- 6.5. Serviços e injeção de dependências
  - 6.5.1. Serviços
  - 6.5.2. Injeção de dependências
  - 6.5.3. *Service Providers*
- 6.6. *Routing* e navegação
  - 6.6.1. Aplicação com *Routing*
  - 6.6.2. *Routing* básico
  - 6.6.3. Rotas aninhadas
  - 6.6.4. Parâmetros
  - 6.6.5. Acesso e autorização
  - 6.6.6. *Lazy Loading* de módulos

- 6.7. RxJS
  - 6.7.1. Observáveis
  - 6.7.2. *Observers*
  - 6.7.3. Subscrições
  - 6.7.4. Operadores
- 6.8. Formulários e HTTP
  - 6.8.1. Formulários reativos
  - 6.8.2. Validação de campos
  - 6.8.3. Formulários dinâmicos
  - 6.8.4. Petições
  - 6.8.5. *Interceptors*
  - 6.8.6. Segurança
- 6.9. Animações
  - 6.9.1. Transações e *Triggers*
  - 6.9.2. Transições de rotas
  - 6.9.3. Diferenças entre transições
- 6.10. *Testing* no Framework Angular
  - 6.10.1. Teste de serviços
  - 6.10.2. Teste de componentes
  - 6.10.3. Teste de diretivas e *pipes*

## Módulo 7. Programação em linguagem NodeJS

- 7.1. NodeJS e a sua arquitetura
  - 7.1.1. NPM e gestão de pacotes
  - 7.1.2. Execução de um programa
  - 7.1.3. Módulos
  - 7.1.4. Criação de um módulo
  - 7.1.5. *Loop* de eventos
- 7.2. Servidor Backend, HTTP, *Express* e *Sockets*
  - 7.2.1. Módulo HTTP
  - 7.2.2. *Express*
  - 7.2.3. *Socket.io*

- 7.3. Bases de dados e caché
  - 7.3.1. MongoDB
  - 7.3.2. Mongoose
  - 7.3.3. SQL
  - 7.3.4. Sequelize
  - 7.3.5. Redis
- 7.4. Sistema de arquivos e Os
  - 7.4.1. *File System Module*
  - 7.4.2. *Os Module*
  - 7.4.3. *Cluster Module*
- 7.5. Eventos, *Buffers* e *Streams*
  - 7.5.1. Eventos
  - 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. Arquitetura e boas práticas
  - 7.7.1. DRY
  - 7.7.2. SOLID
  - 7.7.3. CRUD
  - 7.7.4. MVC
  - 7.7.5. Monolitos
  - 7.7.6. Microserviços
  - 7.7.7. Arquiteturas hexagonais
- 7.8. Typescript
  - 7.8.1. Tipos, interfaces e classes
  - 7.8.2. Funções e módulos
  - 7.8.3. Genéricos
  - 7.8.4. *Namespaces*
  - 7.8.5. Decoradores

- 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. Construção de uma API REST com Express
- 7.10. Criação e contentorização de uma aplicação com NestJS
  - 7.10.1. Nest CLI
  - 7.10.2. Docker
  - 7.10.3. Construção de uma aplicação

## Módulo 8. Bases de dados para programadores *Full Stack*

- 8.1. Bases de dados para programadores *Full Stack*
  - 8.1.1. Base de dados no desenvolvimento de aplicações
  - 8.1.2. Capacidades de bases de dados
  - 8.1.3. SQL (*Structured Query Language*)
- 8.2. Escolha de base de dados
  - 8.2.1. Aplicação ou serviço a considerar
  - 8.2.2. Categorias de bases de dados
  - 8.2.3. Panorama de base de dados
- 8.3. Desenvolvimento com MySQL
  - 8.3.1. Desenvolvimento com MySQL
  - 8.3.2. Implementação de modelo relacional com MySQL
  - 8.3.3. Conexão ao MySQL
- 8.4. Desenvolvimento com Oracle Database
  - 8.4.1. Desenvolvimento com Oracle DB
  - 8.4.2. Implementação do modelo
  - 8.4.3. Conexão a Oracle Database
- 8.5. Desenvolvimento com Oracle SQL Server
  - 8.5.1. Oracle SQL Server
  - 8.5.2. Implementação do modelo
  - 8.5.3. Conexão ao SQL Server

- 8.6. Desenvolvimento com NoSQL
    - 8.6.1. Comparação com bases de dados SQL
    - 8.6.2. Criação de base de dados em MongoDB
    - 8.6.3. Conexão a MongoDB
  - 8.7. Desenvolvimento com grafos
    - 8.7.1. Desenvolvimento com grafos
    - 8.7.2. Criação de base de dados com Neo4j
    - 8.7.3. Conexão com Neo4j
  - 8.8. Desenvolvimento com bases de dados chave-valor
    - 8.8.1. Desenvolvimento com bases de dados k-v
    - 8.8.2. Criação de base de dados com Redis
    - 8.8.3. Conexão com Redis
  - 8.9. Bases de dados com outros tipos de dados
    - 8.9.1. *Elastic Search*
    - 8.9.2. *Inmemory Database*
    - 8.9.3. Desenvolvimento com dados espaciais
  - 8.10. Base de dados. Aspectos avançados
    - 8.10.1. Bases de dados em desenvolvimentos cloud native
    - 8.10.2. Bases de dados em arquitetura de microsserviços
    - 8.10.3. CI/ CD e as bases de dados
- Módulo 9. UX CX. Experiência do cliente**
- 9.1. *HR Customer Experience*
    - 9.1.1. *Customer Experience (CX)*
    - 9.1.2. Novas necessidades no consumo
    - 9.1.3. *Feedback em Customer Experience*
  - 9.2. Tecnologias inovadoras
    - 9.2.1. Máquinas que pensam
    - 9.2.2. Novas formas de partilhar informação
    - 9.2.3. Medição do que não pode ser medido
  - 9.3. Canais de interação com o utilizador
    - 9.3.1. Análise do cliente
    - 9.3.2. Personalização
    - 9.3.3. Múltiplos canais de interação com o utilizador
  - 9.4. Análise de utilizador
    - 9.4.1. Estrutura da web
    - 9.4.2. Análise de utilizador
    - 9.4.3. Análise de utilizador avançado
  - 9.5. Nielsen e o seu impacto na CX
    - 9.5.1. Nielsen e o seu impacto na CX
    - 9.5.2. Técnicas de *User Testing*
  - 9.6. Ferramentas de *Customer Experience*
    - 9.6.1. Ferramentas avançadas
    - 9.6.2. Mobilidade
    - 9.6.3. Acessibilidade
  - 9.7. Novas metodologias
    - 9.7.1. O desafio do utilizador
    - 9.7.2. Processo de UX
    - 9.7.3. Investigação de utilizador
  - 9.8. Comunicação de um desenho
    - 9.8.1. *Wireframing*
    - 9.8.2. Ferramentas de comunicação de um desenho
    - 9.8.3. Ferramentas avançadas de comunicação de um desenho
  - 9.9. Design da UI
    - 9.9.1. Design da UI
    - 9.9.2. Interfaces web e móveis
    - 9.9.3. Componentes web e móveis
  - 9.10. Elaboração de uma CX
    - 9.10.1. Elaboração de uma CX
    - 9.10.2. Elaboração de novas experiências
    - 9.10.3. Interfaces



## Módulo 10. Integração contínua e implantação de aplicações

- 10.1. Integração contínua e implantação contínua: CI/CD
  - 10.1.1. Uso de Integração contínua e implantação contínua (CI/CD)
  - 10.1.2. Diferenças entre Integração contínua e implantação contínua (CI/CD)
  - 10.1.3. Integração contínua e implantação contínua. Benefícios de CI/CD
- 10.2. Os novos paradigmas de desenvolvimento
  - 10.2.1. De Waterfall a DevOps
  - 10.2.2. Guia de estilo: os 12 fatores
  - 10.2.3. Cloud Native, microsserviços e *Serverless*
- 10.3. DevOps, para além de CI/CD
  - 10.3.1. DevOps.
  - 10.3.2. DevOps. *Continuous Everything*
  - 10.3.3. DevOps vs SRE
- 10.4. Tecnologia de contentores I - Docker
  - 10.4.1. Os contentores. Contribuição
  - 10.4.2. Docker. Arquitetura
  - 10.4.3. Processo de implantação com Docker
- 10.5. Tecnologia de contentores II - Kubernetes
  - 10.5.1. Orquestração
  - 10.5.2. Kubernetes
  - 10.5.3. O ecossistema de Kubernetes
- 10.6. Configuração de infraestruturas com GitOps
  - 10.6.1. Infraestrutura imutável
  - 10.6.2. GitOps
  - 10.6.3. Ferramentas GitOps
- 10.7. Pipelines e automatização. Casos de Utilização da CI/CD
  - 10.7.1. Integração contínua
  - 10.7.2. Implementação e entrega contínua
  - 10.7.3. Validações automáticas
  - 10.7.4. Boas práticas em CI/CD
- 10.8. CI/CD com Jenkins. Referência
  - 10.8.1. CI/CD com Jenkins
  - 10.8.2. Pipelines Jenkins
  - 10.8.3. Boas práticas com Jenkins
- 10.9. Ecossistema CI/CD
  - 10.9.1. Organização do ecossistema
  - 10.9.2. Ferramentas avançadas
  - 10.9.3. Dagger. O futuro
- 10.10. Fases finais do ciclo do software orientado para CI/CD
  - 10.10.1. Aplicação de AI ao processo CI/CD
  - 10.10.2. DevSecOps
  - 10.10.3. *Chaos Engineering*



*Terá à sua disposição os mais modernos recursos educativos, com livre acesso à sala de aula virtual 24 horas por dia”*

06

# Metodologia

Este programa de capacitação oferece uma forma diferente de aprendizagem.

A nossa metodologia é desenvolvida através de um modo de aprendizagem

cíclico: **o Relearning.**

Este sistema de ensino é utilizado, por exemplo, nas escolas médicas mais prestigiadas

do mundo e tem sido considerado um dos mais eficazes pelas principais publicações,

tais como a ***New England Journal of Medicine.***



“

*Descubra o Relearning, um sistema que abandona a aprendizagem linear convencional para o levar através de sistemas de ensino cíclicos: uma forma de aprendizagem que provou ser extremamente eficaz, especialmente em disciplinas que requerem memorização"*

## Estudo de Caso para contextualizar todo o conteúdo

O nosso programa oferece um método revolucionário de desenvolvimento de competências e conhecimentos. O nosso objetivo é reforçar as competências num contexto de mudança, competitivo e altamente exigente.

“

*Com a TECH pode experimentar uma forma de aprendizagem que abala as fundações das universidades tradicionais de todo o mundo”*



*Terá acesso a um sistema de aprendizagem baseado na repetição, com ensino natural e progressivo ao longo de todo o programa de estudos.*



## Um método de aprendizagem inovador e diferente

Este programa da TECH é um programa de ensino intensivo, criado de raiz, que propõe os desafios e decisões mais exigentes neste campo, tanto a nível nacional como internacional. Graças a esta metodologia, o crescimento pessoal e profissional é impulsionado, dando um passo decisivo para o sucesso. O método do caso, a técnica que constitui a base deste conteúdo, assegura que a realidade económica, social e profissional mais atual é seguida.

“

*O nosso programa prepara-o para enfrentar novos desafios em ambientes incertos e alcançar o sucesso na sua carreira”*

*O estudante aprenderá, através de atividades de colaboração e casos reais, a resolução de situações complexas em ambientes empresariais reais.*

O método do caso tem sido o sistema de aprendizagem mais amplamente utilizado nas principais escolas de informática do mundo desde que existem. Desenvolvido em 1912 para que os estudantes de direito não só aprendessem o direito com base no conteúdo teórico, o método do caso consistia em apresentar-lhes situações verdadeiramente complexas, a fim de tomarem decisões informadas e valorizarem juízos sobre a forma de as resolver. Em 1924 foi estabelecido como um método de ensino padrão em Harvard.

Numa dada situação, o que deve fazer um profissional? Esta é a questão que enfrentamos no método do caso, um método de aprendizagem orientado para a ação. Ao longo do programa, os estudantes serão confrontados com múltiplos casos da vida real. Terão de integrar todo o seu conhecimento, investigar, argumentar e defender as suas ideias e decisões.

## Relearning Methodology

A TECH combina eficazmente a metodologia do Estudo de Caso com um sistema de aprendizagem 100% online baseado na repetição, que combina elementos didáticos diferentes em cada lição.

Melhoramos o Estudo de Caso com o melhor método de ensino 100% online: o Relearning.

*Em 2019 obtivemos os melhores resultados de aprendizagem de todas as universidades online do mundo.*

Na TECH aprende- com uma metodologia de vanguarda concebida para formar os gestores do futuro. Este método, na vanguarda da pedagogia mundial, chama-se Relearning.

A nossa universidade é a única universidade de língua espanhola licenciada para utilizar este método de sucesso. Em 2019, conseguimos melhorar os níveis globais de satisfação dos nossos estudantes (qualidade de ensino, qualidade dos materiais, estrutura dos cursos, objetivos...) no que diz respeito aos indicadores da melhor universidade online do mundo.



No nosso programa, a aprendizagem não é um processo linear, mas acontece numa espiral (aprender, desaprender, esquecer e reaprender). Portanto, cada um destes elementos é combinado de forma concêntrica. Esta metodologia formou mais de 650.000 licenciados com sucesso sem precedentes em áreas tão diversas como a bioquímica, genética, cirurgia, direito internacional, capacidades de gestão, ciência do desporto, filosofia, direito, engenharia, jornalismo, história, mercados e instrumentos financeiros. Tudo isto num ambiente altamente exigente, com um corpo estudantil universitário com um elevado perfil socioeconómico e uma idade média de 43,5 anos.

*O Relearning permitir-lhe-á aprender com menos esforço e mais desempenho, envolvendo-o mais na sua capacitação, desenvolvendo um espírito crítico, defendendo argumentos e opiniões contrastantes: uma equação direta ao sucesso.*

A partir das últimas provas científicas no campo da neurociência, não só sabemos como organizar informação, ideias, imagens e memórias, mas sabemos que o lugar e o contexto em que aprendemos algo é fundamental para a nossa capacidade de o recordar e armazenar no hipocampo, para o reter na nossa memória a longo prazo.

Desta forma, e no que se chama Neurocognitive context-dependent e-learning, os diferentes elementos do nosso programa estão ligados ao contexto em que o participante desenvolve a sua prática profissional.



Este programa oferece o melhor material educativo, cuidadosamente preparado para profissionais:



#### Material de estudo

Todos os conteúdos didáticos são criados pelos especialistas que irão ensinar o curso, especificamente para o curso, para que o desenvolvimento didático seja realmente específico e concreto.

Estes conteúdos são depois aplicados ao formato audiovisual, para criar o método de trabalho online da TECH. Tudo isto, com as mais recentes técnicas que oferecem peças de alta-qualidade em cada um dos materiais que são colocados à disposição do aluno.



#### Masterclasses

Existem provas científicas sobre a utilidade da observação por terceiros especializada.

O denominado Learning from an Expert constrói conhecimento e memória, e gera confiança em futuras decisões difíceis.



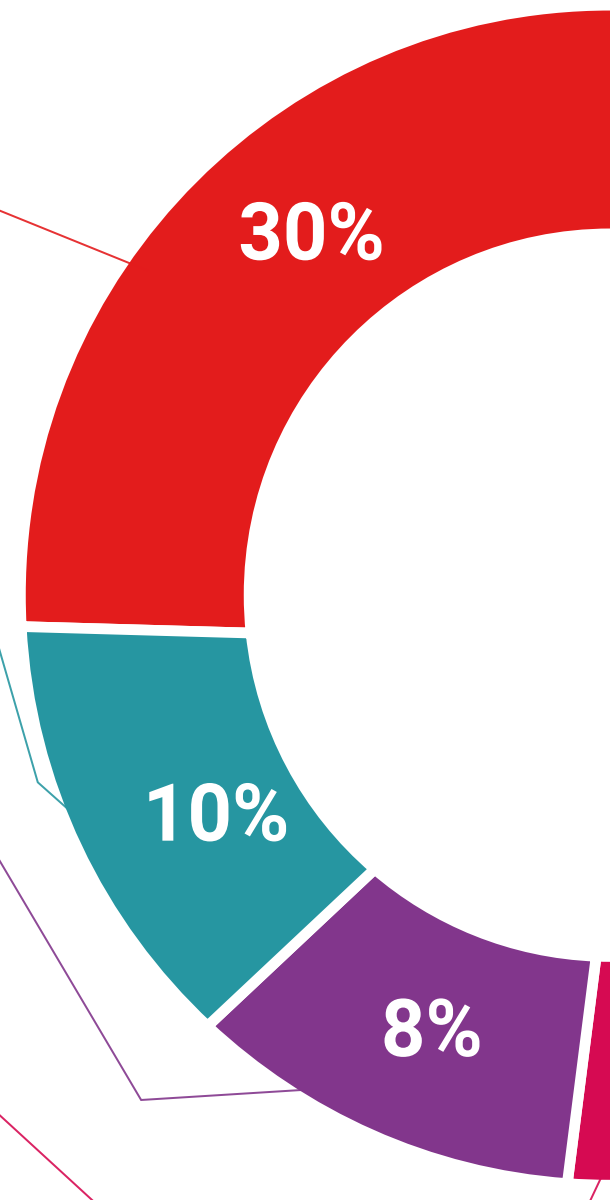
#### Práticas de aptidões e competências

Realizarão atividades para desenvolver competências e aptidões específicas em cada área temática. Práticas e dinâmicas para adquirir e desenvolver as competências e capacidades que um especialista necessita de desenvolver no quadro da globalização em que vivemos.



#### Leituras complementares

Artigos recentes, documentos de consenso e diretrizes internacionais, entre outros. Na biblioteca virtual da TECH o aluno terá acesso a tudo o que necessita para completar a sua capacitação







#### Case studies

Completarão uma seleção dos melhores estudos de casos escolhidos especificamente para esta situação. Casos apresentados, analisados e instruídos pelos melhores especialistas na cena internacional.



#### Resumos interativos

A equipa da TECH apresenta os conteúdos de uma forma atrativa e dinâmica em comprimidos multimédia que incluem áudios, vídeos, imagens, diagramas e mapas conceituais a fim de reforçar o conhecimento.

Este sistema educativo único para a apresentação de conteúdos multimédia foi premiado pela Microsoft como uma "História de Sucesso Europeu"



#### Testing & Retesting

Os conhecimentos do aluno são periodicamente avaliados e reavaliados ao longo de todo o programa, através de atividades e exercícios de avaliação e auto-avaliação, para que o aluno possa verificar como está a atingir os seus objetivos.



07

# Certificação

O Mestrado Próprio em Programação Front End Full Stack Developer garante, para além de um conteúdo mais rigoroso e atualizado, o acesso a um grau de Mestre emitido pela TECH Universidade Tecnológica.



“

*Conclua este plano de estudos com sucesso e receba o seu certificado sem sair de casa e sem burocracias”*

Este **Mestrado Próprio em Programação Front End Full Stack Developer** conta com o conteúdo educacional mais completo e atualizado do mercado.

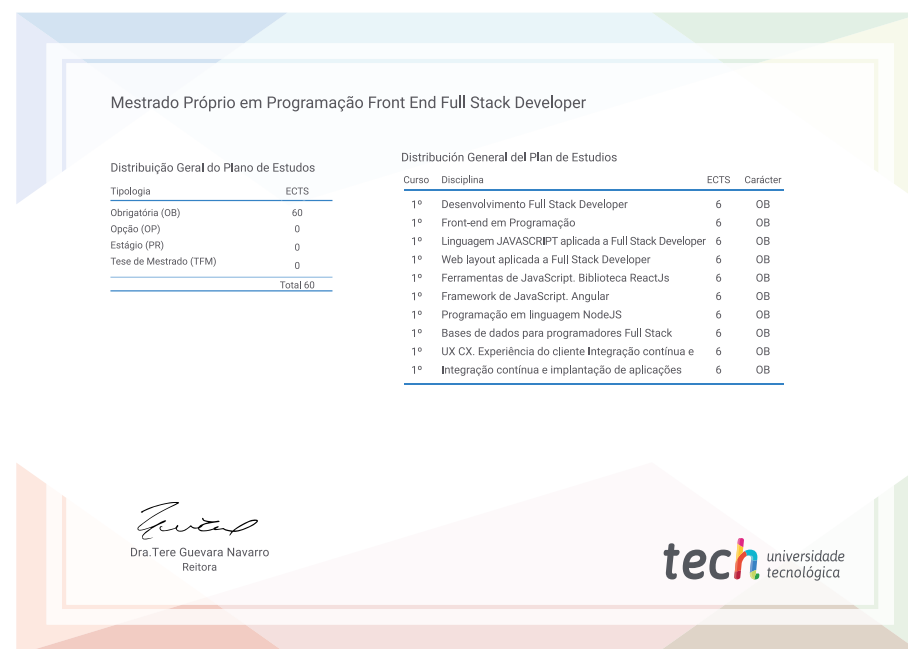
Uma vez aprovadas as avaliações, o aluno receberá por correio, com aviso de receção, o certificado\* correspondente ao título de **Mestrado Próprio** emitido pela **TECH Universidade Tecnológica**.

O certificado emitido pela **TECH Universidade Tecnológica** expressará a qualificação obtida no Mestrado Próprio, atendendo aos requisitos normalmente exigidos pelas bolsas de emprego, concursos públicos e avaliação de carreiras profissionais.

Certificação: **Mestrado Próprio em Programação Front End Full Stack Developer**

ECTS: **60**

Carga horária: **1500 horas**



\*Apostila de Haia Caso o aluno solicite que o seu certificado seja apostilado, a TECH EDUCATION providenciará a obtenção do mesmo com um custo adicional.



## Mestrado Próprio

Programação Front End Full Stack Developer

- » Modalidade: online
- » Duração: 12 meses
- » Certificação: TECH Universidade Tecnológica
- » Créditos: 60 ECTS
- » Tempo Dedicado: 16 horas/semana
- » Horário: ao seu próprio ritmo
- » Exames: online

# Mestrado Próprio

## Programação Front End Full Stack Developer