



## FLAGProfessional Full-Stack Web Developer

Desenvolvimento

Live Training ( também disponível em presencial )

- **Localidade:** Lisboa
- **Data:** 16 Nov 2022
- **Preço:** 3750 € ( Os valores apresentados não incluem IVA. Oferta de IVA a particulares e estudantes. )
- **Horário:** Pós-Laboral das 2<sup>as</sup>, 4<sup>as</sup> e 6<sup>as</sup>, das 18h30 às 21h30
- **Nível:** Entrada
- **Duração:** 360h

---

### Sobre o curso

Os Developers já não desempenham um papel único no desenvolvimento, tornaram-se multifacetados, assumindo o papel de designers, developers, coders e especialistas em bases de dados, e assim por diante. O conceito de Full Stack Developer não significa um profissional que sabe fazer tudo.

Especificamente, o termo e função refere-se a um perfil que domina aptidões multi-tasking e pode usá-las para completar de forma independente um projeto ou desenvolvimento de um produto digital. Este pensamento mais abrangente é a maior vantagem de ser um Full Stack Developer, pois este estará mais familiarizado com todas as fases e processo de desenvolvimento e design do produto, e mesmo de Marketing. E isso tornará a cooperação dentro e fora da equipa mais otimizada, reduzirá os custos e os tempos de produção do projeto e precaverá situações inesperadas.

#### Saídas profissionais:

- Back-end Developer
- Front-end Developer
- Web Developer
- Full-stack Developer

---

### Destinatários

Todos os interessados em iniciar ou consolidar uma atividade profissional como Full Stack Developer.

Note-se que, os cursos FLAGProfessional desenvolvem-se com graus de exigência extremamente

elevados – tanto em carga horária como pela parte prática -, o que obriga a uma total disponibilidade, do formando, para o correto acompanhamento das matérias leccionadas, bem como dos projetos a realizar.

---

## Objetivos

O curso Full Stack Developer, pretende dotar os formandos de todos os conhecimentos teóricos e competências necessários para ingressar no mercado de trabalho e assumir uma atividade profissional ligada a competências com tecnologias de back-end e front-end. Serão criadas bases sólidas para trabalhar com servidores e configurações de host, executando integrações de bases de dados e solucionando problemas de desenvolvimento de Front-end.

---

## Condições

- Taxa de inscrição: 10% do valor do curso, dedutível no valor total
  - Possibilidade de pagamento faseado do restante valor para particulares
  - **100€ de Desconto** para inscrições a pronto pagamento
  - Estudantes não residentes no território nacional, terão de efetuar um pagamento de 50% do valor total da propina no momento da inscrição.
  - Os valores apresentados não incluem IVA. Isenção do valor do IVA a particulares.
  - Para informações completas sobre os requisitos e condições financeiras disponíveis, contacte-nos através do botão Saber +
- 

## Pré-requisitos

Gosto e apetência pela área de desenvolvimento, não existem pré-requisitos técnicos muito específicos para a inscrição neste curso, o programa foi projetado e desenvolvido para englobar vários perfis de alunos.

---

## Metodologia

Formação Presencial ou Live Training (Formação Online Síncrona).

---

# Programa

- Fundamentos de Web Development (3h)
- Metodologias Agile para Gestão de Projetos (9h)
- Workshop: Scrum (9h)
- Fundamentos de UI/UX para Developers (9h)
- HTML5 (21h)
- CSS3 (27h)
- Workshop: SASS (9h)
- Workshop: Responsive Web Design (12h)
- Fundamentos de Programação (com Javascript) (15h)
- Javascript (54h)
- Frontend frameworks – Angular (18h)
- Frontend frameworks – React (18h)
- Workshop: GIT (9h)
- Projeto (18h)
- Fundamentos de Back-End Development (3h)
- Introdução à estruturação de dados (9h)
- Bases de dados: MySQL (12h)
- PHP (30h)
- Workshop: PHP Frameworks (21h)
- Criação de REST API'S (12h)
- Workshop: MongoDB + NodeJS + ExpressJS (24h)
- Projeto (18h)

## FUNDAMENTOS DE WEB DEVELOPMENT

Protocolos de comunicação na web

Tecnologias e ferramentas front-end e back-end

Gestão de projectos web

Domínios e alojamento de sites

## METODOLOGIAS AGILE PARA GESTÃO DE PROJETOS

Metodologias Processo de Design Digital

Processos de desenvolvimento de produtos (Agile, Lean e Waterfall)

Conceito de MVP (Minimum Viable Product)

Processo de design (etapas)

The Design Sprint (Google Product Design Sprint)

Manifesto Ágil, Valores e Princípios Ágeis

#### **WORKSHOP: SCRUM**

Manifesto Ágil

Raízes do SCRUM

Visão geral Framework SCRUM

Papéis e Responsabilidades: PO, SM e Time

Artefatos do SCRUM

Eventos

Implementação

Métricas

#### **FUNDAMENTOS DE UI/UX PARA DEVELOPERS**

Introdução ao Ux & Ui

Multi-screen & Responsive Challenges

UX/UI and development

Arquitetura de informação

Design Patterns & Componentes

Wireframing & Prototipagem

Assets for development

#### **HTML5**

Estrutura de uma página HTML

Elementos e atributos

Elementos de texto

Listas

Hiperligações

Imagens

Tabelas

Formulários

Elementos multimédia

Elementos estruturais do HTML5

**CSS3**

Evolução das CSS

Utilizar CSS em páginas Web

Selectores CSS

Unidades

Trabalhar com cores

Vendor prefixes

Formatação de texto

Tipografia

Backgrounds e gradientes

Borders e sombras

Box model

Layout de páginas

Novos métodos de layout em CSS3

Propriedades para tabelas e formulários

Criação de elementos de navegação

CSS sprites e elementos interactivos

Transforms, transitions e animations

Frameworks front-end – Bootstrap

#### **WORKSHOP: SASS**

Preprocessing

Variáveis

Nesting

Partials

Modules

Mixins

Extend & Inheritance

Operadores

#### **WORKSHOP: RESPONSIVE WEB DESIGN**

Princípios básicos do responsive

Responsive websites vs. Mobile website vs. Mobile app

Introdução ao processo de desenvolvimento

Html5 + CSS3 princípios básicos

Media queries

Fix and fluid Layouts

Adaptive and responsive web design

Responsive design

Frameworks

Projeto Prático

#### **FUNDAMENTOS DE PROGRAMAÇÃO (COM JAVASCRIPT)**

Paradigmas de Programação (Object Oriented, Imperative e Functional Programming,...)

Processo de desenvolvimento de um algoritmo (programa) (Utilização de pseudo-código e fluxogramas)

Tipos de Dados numa linguagem de programação (Variáveis, constantes, ...)

Operadores e cálculos numa linguagem de programação

Controle da execução de um programa com recurso a estruturas de decisão e repetição

## JAVASCRIPT

Introdução, história e evolução dos vários standards da linguagem

Breve introdução á historia de desenvolvimento da linguagem, e a sua evolução até aos dias de hoje.

Conceitos / Instruções Básicas em javascript

- Expressões
- Comentários
- Variáveis e constantes
- Operadores

Tipos de dados primitivos em Javascript

- Numbers, Strings, Boolean, null, undefined

Estruturas de decisão e repetição em javascript

- Estruturas de decisão no controle do fluxo de um programa: (If, if/else, switch)
- Estruturas de repetição: (For, While, Do/While)

Tipos de dados Complexos, Arrays, Objectos, Funções e Built-In Objects

- Utilização de Arrays
- Criação de Objectos, propriedades e métodos
- Conceito de Função, funções com parâmetros
- Funções anónimas e IIFE (Immediate Invoked Function Expression)
- Funções “construtoras” de Objectos
- Conceito de “Scope” de uma variável, variáveis locais e globais, diferenças em ES6 com let e const
- Built-In Objects em JS

Browser Object Model (BOM)

Document Object Model (DOM)

Global Objects (Number, String, Boolean, Undefined, Null, Object, Math, Date, ...)

Conceito e manipulação do DOM (Document Object Model) numa aplicação web

- Como o Browser interpreta o HTML e cria um Objecto (DOM), representativo dos elementos da página
- Tipos de elementos(Nodes) do DOM (Document, Element, Attributes e TextNodes)
- Processo e Conceito de manipulação do DOM
- Metodos para seleccionar um elemento(node) ou uma coleção de elementos(nodelist) do DOM (Live e static methods)
- Propriedades e métodos para percorrer o DOM
- Alterar conteúdos de elementos do DOM
- Inserir, mover ou eliminar elementos do DOM
- Manipulação de atributos de elementos do DOM
- Trabalhar com Eventos em Javascript

Conceito de interactividade com Eventos em JS

Categorias (tipos) de Eventos em JS

Definição de Eventos em JS (HTML event handler's, Event Handler's, DOM EventListeners)

Conceito de Event Flow e suas fases em JS (Capturing, targeting e bubbling)

Conceito de Event Object em JS

Delegação de Eventos em JS

Introdução á especificação ES6

Evolução do standard de javascript e novas especificações ES6, ES7 e ES8

Utilização de “Transpilers” como o BabelJS para a compilação de ES6 em “Vanilla Javascript” standard (ES5)

Variáveis e parâmetros em ES6 (let e const, Destructuring, Default Parameters, Classes, Rest e Spread operators, multiline e template strings,...)

Arrow functions, Iterators, e outros conceitos de Functional programming

Novos métodos de manipulação de Arrays e Objectos (map, reduce, Object.Assign(), ...)

AJAX e JSON com Javascript (ES6)

- Conceito de Ajax com JS (requests e responses)
- Tipos de dados numa comunicação assíncrona com o servidor
- Conceito de Promises (ES6)
- Utilização da “Fetch API” para chamadas assíncronas.
- Gestão de erros em Javascript



- Filtrar, Ordenar e pesquisar numa estrutura de dados em JS
- Introdução e utilização de algumas API's de HTML5 (LocalStorage, SessionStorage, Geolocation, ...)

## FRONTEND FRAMEWORKS – ANGULAR

### Introdução, Ambiente de Desenvolvimento:

- Breve introdução ao ECMAScript, novas funcionalidades da especificação ES6 (Classes, fat Arrow functions, spread e rest operators, destructuring, ...)
- Breve introdução á sintaxe do Typescript
- Gestão de “packages, bibliotecas e frameworks” de Javascript com npm (ou yarn)

### Introdução ao Angular

- Enquadramento da utilização da framework Angular para desenvolvimento Web
- Abordagem e explicação genérica sobre a framework
- Utilização de Angular CLI como ferramenta para gerir, e desenvolver projetos em Angular
- Conceitos de compilação JIT (Just in Time) e AoT (ahead of time)
- Demonstração prática da estrutura de uma aplicação em Angular

### Elementos estruturais de uma Aplicação em Angular

- Components
- Services
- Directives
- Pipes
- Modules
- Conceito de Data-Binding em Angular

### Navegação entre componentes de uma Aplicação – Routing

- Conceitos básicos de “Routing”
- Implementação de “Routing” e módulos que compõem uma estrutura de navegação
- Passagem de “Data” em “Routing”
- Conceito de “Child Routes”

### Dependency Injection em Angular

- Introdução ao Padrão de Desenho : “Dependency Injection”
- “Injectors” e “Providers”, “injectar” um serviço, utilização do HttpClient.
- Reactive programming em Angular, com Observables

### Estratégia de comunicação entre componentes

- Criar “loosely-coupled” components (@Input e @output properties)
- Como passar dados entre componentes hierarquicamente relacionadas
- Utilização do Padrão de Desenho “Mediator” através de DI de serviços
- “Change Detection” em Angular e métodos “lifecycle” de uma componente

## Introdução a API Forms do Angular

- Template-driven forms em Angular
- Reactive forms em Angular
- Utilizar o FormBuilder
- Validação de Forms com Validators Built-in

## Interacção com servidores através do HttpClient

- Introdução ao Modulo HttpClient
- Interacção com webserver em “node” e Typescript
- Efetuar chamadas para o servidor
- Reactive Extensions (NGRX) em Angular

## FRONTEND FRAMEWORKS – REACT

### Build Workflow

### Componentes

### JSX

### Restrições JSX

### State

### Handling Events

### Two Way Binding

### Lists, keys & mapping

### Components Lifecycle

### React Hooks

### Ajax in React

### Axios

### Forms and Form Validation

Redux

Redux Advanced

Deploying React project

NEXT.js

#### **WORKSHOP: GIT**

Instalação e configuração de Servidor GIT

Instalação de Cliente GIT

Criar uma equipa

Criar um projecto

Configurar repositório

Criar um branch a partir de master

Commit

Revert

Merge

Deploy

#### **PROJETO**

#### **FUNDAMENTOS DE BACK-END DEVELOPMENT**

Modelo Cliente – Servidor

Protocolos de transferência de informação

Configuração da máquina de desenvolvimento

#### **INTRODUÇÃO À ESTRUTURAÇÃO DE DADOS**

Modelo Relacional

Entidades ; atributos; relacionamentos

Tipos de relação

Formas normais

**BASES DE DADOS: MYSQL**

Modelo físico

Construção de queries MySQL: INSERT, DELETE, CREATE, SELECT, UPDATE, DROP, JOIN,...

SP e Cursors

**PHP**

Introdução ao PHP

- Anatomia de um script
- Tipos de dados
- Variáveis e operadores
- Estruturas de controlo e decisão
- Gestão de erros

Funções

- Sintaxe base
- Retornar valores
- Domínio de variáveis (scope)

Arrays

- Introdução às arrays
- Operações (comparação, contagem, pesquisa e gestão de elementos)
- Iterações (ponteiros)
- Ordenação

Strings

- Introdução às strings
- Comparação, pesquisa e substituição de caracteres
- Formatação

OOP

- Declarar uma classe e criar instâncias
- Métodos e propriedades
- Interfaces

## Programação Web

- Formulários e URLs (GET, POST, Uploads)
- HTTP Headers (redireccionamento, cookies, sessões)
- Envio de emails (função mail)

## Integração com MySQL

- Abrir e fechar uma ligação ao MySQL
- Executar comandos SQL
- Processar dados da base de dados

## Streams

- Aceder a ficheiros

## Trabalhar com diretorias

### **WORKSHOP: PHP FRAMEWORKS**

O que é uma framework e uma micro-framework

Vantagens de usar uma Framework

Diferenças entre frameworks

Utilização do composer

Padrão de desenho MVC

O que é o controlador

O que é o modelo

O que é uma vista

Criação de uma aplicação PHP usando uma Framework

### **CRIAÇÃO DE REST API'S**

O que é uma API REST

Protocolo HTTP

Tipos de pedidos e códigos de resposta

Respostas JSON

## Respostas XML

Autenticação com base em tokens

Criação da documentação para a API

### **WORKSHOP: MONGODB + NODEJS + EXPRESSJS**

#### MongoDB

- Instalação
- Criação de modelo de dados
- Querys

#### NodeJS

- O que é o NodeJS
- Utilização do NPM – Node Package Manager
- Criação de uma aplicação NodeJS
- Funcionamento do express para resposta a pedidos HTTP
- Bases de dados não relacionais com MongoDB
- Utilização de WebSockets usando SocketIO

#### NativeScript

- O que é o NativeScript
- Como funciona o NativeScript
- Instalação das ferramentas de desenvolvimento
- Criação de UI
- Arquitetura da aplicação e navegação
- Módulos
- Data binding
- Eventos
- Integração com API's nativas

### **PROJETO**