Curso: Academia FLAGProfessional Full Stack Web Development

Duração: 378h
Área formativa: Cursos

Sobre o curso

Curso Profissional de Full Stack Web Development

Os programadores, ou developers, têm sido dos profissionais mais procurados no mercado de trabalho, com salários acima da média.

A elevada empregabilidade, associada aos desafios tecnológicos com que as organizações se deparam, tornaram a carreira de Full Stack Web Developer muito apelativa, e de acordo com Future of Jobs Report 2025 do World Economic Forum, a procura por Full Stack Developers deverá crescer cerca de 25% até 2030.

Enquanto full stack web developer irás apresentar aptidões multi-tasking no que diz respeito ao desenvolvimento de produtos digitais. Serás assim um elemento-chave na cooperação e otimização de equipas, ajudando na redução de custos e tempos de produção, e estarás mais capacitado a identificar e prevenir situações inesperadas.

Porquê apostar numa carreira em Programação Web?

#Empregabilidade O mercado está cheio de ofertas de emprego para developers de soluções web nos mais variados domínios e a possibilidade de emprego além-fronteiras é uma realidade.

#Versatilidade Na área de desenvolvimento web tens variadas opções de carreira, ditadas pela linguagem de programação em que te especializares ou meio para o qual programares.

#Valorização Ao complementares a tua formação de base com conhecimentos de programação estarás a valorizar o teu currículo e percurso formativo, com uma forte competência digital, que te irá distinguir enquanto profissional com visão tecnológica.

#Reconhecimento De acordo com o relatório Tech Careers 2020 da Landing.jobs, o salário médio do perfil de programador aumentou mais do que 15% nos últimos três anos.

Por onde deves começar?

Com a **Academia FLAGProfessional Full Stack Web Development** vais:

:: **Dominar todos os conceitos-base** necessários ao trabalho com servidores, configurações de host e frameworks modernas (React, Node.js), integrações de dados (MySQL) e solução de problemas de back e front-end;

:: **Aprender a trabalhar com metodologias ágeis** (Scrum), aplicando-as ao longo de projetos reais e preparados para o mercado;

:: **Conhecer as melhores práticas no mercado** da programação web, a nível global, incluindo DevOps e integração contínua, para assegurar a eficiência e escalabilidade dos teus projetos;

:: **Estudar as principais linguagens de programação,** frameworks e interfaces/softwares usados no desenvolvimento web: HTML, JavaScript, PHP, CSS, GIT, REST APIs e ferramentas de automação.

:: **Construir o teu portfólio** profissional através dos projetos desenvolvidos ao longo do curso, para que possas demonstrar o teu know-how, criatividade e capacidade de execução às entidades empregadoras.

:: Ter acesso ao **Employability Hub**, um serviço dedicado a apoiar a integração e a progressão de carreira dos formandos das Academias da FLAG. Oferecemos um acompanhamento personalizado, focado na maximização do teu posicionamento no mercado de trabalho. Descobre mais sobre o Employability Hub aqui.

Este curso pode ser totalmente adaptado em conteúdos, duração, datas e horários às tuas necessidades pessoais ou para um grupo de colaboradores da tua empresa. Queres saber mais sobre as nossas soluções de formação à medida?

Objectivos

Ao terminares este curso poderás trabalhar em qualquer empresa que desenvolva (para os seus clientes internos ou externos) produtos web, desempenhando funções de:

:: **Full stack Web Developer**, enquanto especialista que domina aptidões multi-tasking quer do lado do cliente quer do lado do servidor, e está familiarizado com as várias tecnologias envolvidas na programação e design de websites.

:: Front-end Developer, enquanto profissional responsável por garantir que o front-end visual de um site esteja livre de erros e se pareça exatamente com o que foi projetado, e que tenha a mesma visibilidade em diferentes computadores e sistemas operativos.

:: Back-end Developer, enquanto profissional que cria ou contribui para o bom funcionamento de aplicações para a web e mantém um código limpo, leve e bem documentado.

:: Web Developer, enquanto programador especializado no desenvolvimento de sistemas para a internet, responsável pelo perfeito funcionamento de um website ou aplicação web.



Certificado de Formação

Ao concluíres com aproveitamento esta formação, receberás um Certificado de Formação (acreditado pela DGERT) e um badge digital (da Credly) para partilhares com a tua rede profissional online.
Saber mais sobre os badges digitais

Pré-requisitos

Para frequentares a **Academia FLAGProfessional Full Stack Web Development** deverás ter conhecimentos de informática na ótica do utilizador, mas também raciocínio lógico, domínio do inglês, orientação para a resolução de problemas e uma mente criativa e curiosa.

Considerando o elevado grau de exigência das Academias FLAGProfessional é fundamental que sejas organizado, focado e disciplinado, para que consigas garantir o correto acompanhamento das matérias lecionadas, bem como dos projetos a realizar.

Destinatários

Este curso é indicado para ti se:

:: Pretendes iniciar uma carreira na área de programação, e procuras **complementar o currículo com formação prática** que te ajude a **integrar** rapidamente no **mercado de trabalho**;

:: És profissional na área de programação – software ou applications developer – e pretendes **melhorar** o teu **portfólio**, alargar os teus horizontes e **enriquecer** as tuas **competências** com valências na área de **desenvolvimento web**, ou fazer uma **atualização** aos teus **conhecimentos**;

:: És um empreendedor e pretendes **dar forma aos teus projetos**;

:: Procuras oportunidades de **networking com profissionais ativos no mercado da programação web**.

Este curso pode ser totalmente adaptado em conteúdos, duração, datas e horários às tuas necessidades pessoais ou para um grupo de colaboradores da tua empresa.
Queres saber mais sobre as nossas soluções de formação à medida?

Programa

Sessão de Onboarding

Metodologias Ágeis para Gestão de Projetos: Scrum

Introdução à Gestão de Projetos

- Gestão de Projetos tradicional
- Waterfall ou Agile?
- Projeto vs Produto

Agile – o que é

- Origem
- Contexto Histórico
- Porquê Agile?
- Ser vs Fazer
- Cultura

O Agile Manifesto

- Valores
- Princípios
- Relação com Lean e outros

Scrum – introdução

- Contexto
- Pilares, Valores e Princípios

Scrum – o processo

- Roles
- Eventos
- Artefactos
- O processo

Scrum – relação com outras frameworks

- Scrum e Kanban
- Scrum e outras frameworks e ferramentas

Escalar Agile

- Frameworks em Escala
- Qual escolher?

Fundamentos de UX/UI 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

Introdução à Programação e Desenvolvimento Web

Fundamentos de Programação

- O que é programação.
- Programação vs Linguagem
- Conceitos de “ordem”
- Usando o MIT Scratch como ferramenta de aprendizagem

Algoritmos

Algoritmos computacionais

Pseudocódigo

Arquitetura da Web

- O que é a Web: cliente e servidor, como a web funciona
- IP’s, Servidores, DNS
- Servidores e Cloud
- Alojamentos

Introdução ao Front-End e Back-End

- Introdução ao front-end e back-end: conceitos e responsabilidades
- Como funciona o HTML, CSS e JavaScript
- Como funciona o back-end: servidores e linguagens comuns
- Conceitos de base de dados

HTML e CSS

Introdução ao HTML

- O que é o HTML
- Evolução do HTML até ao HTML5
- As 3 linguagens base da Web
- Para que serve o HTML
- Ferramentas Úteis

Sintaxe do HTML

- Escrita do elemento base
- Atributos
- Regras de construção
- Estrutura base de uma página HTML

Elementos de texto e Imagem

- Aprender elementos como títulos
- Hierarquia dos headings
- Parágrafos
- Elementos de formatação de texto (bolds, itálicos etc)
- Criação de imagens

Hiperligações

- Links absolutos vs relativos
- Anchor vs Button
- Atributos do elemento <a>
- Acessibilidade no HTML
- Links com imagens

Listas

- Tipos de listas (ordenadas e não ordenadas)
- Regras de construção das listas
- Menus (listas com links)
- Atributos da lista

Tabelas

- Tag table
- Construção em linhas
- Elementos constituintes da tabela: thead, tbody e tfoot
- Células em múltiplas linhas ou múltiplas colunas

Elementos multimédia

- Formatos de ficheiros. Quais existem e para que servem
- Elemento <video> e <audio>
- Acessibilidade nas imagens, e nos elementos de vídeo e áudio dentro do html

Iframes

- Iframe
- Como funciona um iframe dentro do html
- Exemplos de incorporação de Youtube e Google Maps

Formulários

- Definição form
- Limitação de formulários dentro do html
- Tag <form>
- Tag <input> e os diversos tipos
- Atributo “name”
- Relação das labels com os inputs
- Elemento Textarea
- <select> e <option> dentro de um formulário
- Como submeter um formulário

Elementos Estruturais do HTML

- O elemento básico da , o que é e para que serve
- Elementos de HTML5 estruturais e com semântica (<section>, <nav> etc)
- A acessibilidade dentro de uma página
- A importância da sintaxe

Regras e boas práticas de acessibilidade

- O que é o ARIA
- A importância da acessibilidade
- Como melhorar a acessibilidade com atributos simples de html

Exercício prático de implementação

- Implementação de um layout básico, de forma a cimentar os conhecimentos de HTML

Introdução à CSS

- O que é a CSS
- Regra do “Cascading” na folha de estilos

Utilizar CSS em páginas web

- Formas de adicionar CSS
- Como o CSS afeta a web
- Várias folhas de estilo na mesma página

Seletores CSS

- Sintaxe de um selector e as suas propriedades
- Selectores de Elementos
- Selectores de Classes
- Selectores de ID
- Selector Universal
- Selectores de Atributos
- Selectores compostos/complexos

Especificidades dentro do CSS

- Regra de especificidades entre elementos, classes e IDs
- A Regra da cascata
- O CSS inline
- A regra do !important

Unidades de Medida

- Entender unidades absolutas e relativas
- Conhecer as várias unidades na CSS como px, rem, e vw
- Percentagens dentro da CSS

Cores

- Sistemas de cor na web
- Propriedades que podem receber cor
- Como trabalhar unidades de cor

Trabalhar o texto

- Propriedades afetas ao texto
- A diferença entre trabalhar o aspeto do texto e o significado do elemento de HTML

Tipografia

- Alterar a font do nosso site ou elemento
- Adicionar fonts google no documento
- Adicionar fonts “custom” com o @fontface

Backgrounds

- Trabalhar o background-color
- Trabalhar as background images e todas as propriedades associadas
- Background-attachment
- Como criar gradients
- Shorthand de background

Borders e shadows

- Como funciona as borders no CSS, e que espaço ocupam
- Shorthand para borders
- Como funcionam as shadows
- Trabalhar as shadows de forma eficaz
- Trabalhar cantos arredondados

Box Model

- O conceito de box-model
- A propriedade box-sizing
- Como funciona o padding
- O que é a Margin

Positions em CSS

- Tipos de posicionamento (Absoluto, Relativo, Fixo etc)
- Como as propriedades Top, Bottom, Right e Left são afetadas pelo tipo de posicionamento

Tipos de Displays

- Os vários valores de display no CSS (block, inline, flex, grid etc.)
- Qual a diferença entre Flex e Grid em relação aos outros

Métodos de layout em CSS

- Entender o display flex
- O que são flex-childs
- A Gap
- Propriedades do Flex container
- Propreidades do Flex Child
- A CSS Grid, o que é e porque é diferente do flex
- Como criar uma grid com colunas
- Como posicionar elementos dentro das colunas
- Como ocupar mais de um espaço dentro da minha grid

Pseudo-classes e pseudo-elementos

- O que são Pseudo-classes como o hover
- Os pseudo-elementos ::before e ::after

Overflow e Z-indez

- Entender o que é a propriedade overflow e como pode ser trabalhada na construção de layouts
- O que é a propriedade z-indez, e como é afetada pelo tipo de posicionamento dos elementos

Transforms, Transitions e Animations

- Propriedade de Transform, e os seus diferentes valores
- Usar as transitions para alterações de estado
- As @keyframe animations e como se usam

Workshop: Versões de Código com Github

O que é um Controlador de versões

- Objetivo
- As mais conhecidas filosofias de versionamento de código
- Filosofia do controlo de versões recorrendo a Git
- Vantagens da abordagem GIT

Instalação e uso do software Git

- Instalação do GIT
- Configurações e autenticação
- Controlo de versões de código
- Partilha de informação entre diferentes elementos da equipe

- Repositórios
- Conceito de Branch e casos de utilização

Utilização do Git para controlo de versões de código

- Boas práticas para criação de repositórios
- Instrução add
- Instrução push
- Instrução commit
- Instrução merge
- Resolução de problemas
- Criação e gestão de um novo branch

Conhecer o GitHub

- Criação e configuração de conta no Github
- Criação e Gestão de projetos
- Criação e Gestão de utilizadores e equipas
- Visibilidade de repositórios
- Análise de histórico de projeto
- Controlo de qualidade recorrendo a ferramenta Github

Javascript para Frontend

Introdução, história e evolução dos vários standards da linguagem

Conceito de variável e Tipos de Dados numa linguagem de programação (Variáveis, constantes, …)

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 (Imediate Invoked Function Expression)
- Funções “construtoras” de Objectos
- Conceito de “Scope” de uma variável, varáveis locais e globais, diferenças em ES6 com let e const

- Built-In Objects em JS

Object Oriented, Imperative e Functional Programming e como usá-las em Javascript

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
- Métodos para selecionar 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 à 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, …)

React Foundations

A Biblioteca React.js

- A organização do código tendo em vista os componentes
- Introdução a biblioteca de React,js
- Princípios de arquitetura da biblioteca React.js
- Introdução a organização do código tendo em conta a arquitetura do React.js
- Criação de um projeto React.js
- Organização de um projeto React.js
- Criação de um componente base
- Introdução ao conceito de SASS
- Debug de aplicação React.js

JSX

- O que é o JSX
- Vantagens do uso do JSX
- Estruturar programa para desenvolvimento em JSX

Componentes React.js

- Components and properties
- Componente lifecycle
- Tipos de components
- Props e Classes
- Nested Components
- Handling Data
- Use Effect e React.js ho

React.js

- Uso do contexto em React.js
- Nested Components
- Databind em aplicação React.js
- Passagem de parâmetros entre componentes
- Partilha de contexto em react.js
- Gestão de estado em react.js
- Pedidos assíncronos em React.js
- Forms validation
- React.js Hooks

Organização de código em Single page applications baseadas em React.js

- Introdução as boas práticas de arquitetura e implementação de SPA’s
- Identificação de erros comuns

- Use cases de implementação e arquitetura baseados em cenários reais

Debuging em React.js

- Devtools em React.js
- Debuging de hierarquia de componentes
- Debuging de propriedades
- Debuging de funções em aplicações React.js

React Avançado

Gestão de estado entre componetes react

- Soluções out of the box da biblioteca react
- Soluções baseadas em arquitetura e hierarquia de componentes

Redux

- Introdução ao Redux
- O Redux como meio de comunicação entre componentes
- Implementaçães baseadas em Redux
- Componentes Redux (Store, Action, Reduer e Dispatcher)

Debugging em redux

- Devtools para redux
- Técnicas de debug em soluções baseadas em redux

Pitstop

Projeto: Aplicação interactiva com React

Bases de dados relacionais com MySQL

Implementação de bases de dados usando o modelo relacional

- Conceitos de bases de dados relacionais e como desenhar base de dados, usando diagramas de entidade relação
- Conceitos de tabelas, registos, chaves, graus de relações etc.

Criação de diagramas entidade relação

- Criação de DER, para visualização da base dados

Instalação e configuração do MySQL

- Instalação e interação com MySQL

Linguagem SQL

- Uso da linguagem SQL para interagir com o MySQL

Comandos DDL

- Comandos SQL para criação de estrutura (Data Definition Language) da base dados.
- Comandos como CREATE para criação de estruturas como a própria base dados ou tabelas, comando DROP para apagar estruturas, etc.

Comandos DML

- Comandos SQL para interagir com os dados (Data Manipulation Language).
- Comandos SELECT para obter dados da base de dados e aprender aprofundadamente como este comando funciona (LIMIT, GROUP BY, HAVING, AS, etc.)
- Comando INSERT para inserir dados na base de dados.
- Comando UPDATE para atualizar dados na base de dados.
- Comando DELETE para apagar dados da base de dados.
- Filtrar dados com o comando WHERE e os seus operadores mais complexos (IN, BETWEEN, LIKE, etc.)

Junção de tabelas com JOINS

- Consultar dados que dependem de múltiplas tabelas usando o JOIN
- Diferentes tipos de JOIN e quando usar cada um deles

Subqueries

- Como criar consultas umas dentro de outras, quando é necessário criar um dataset dinâmico

Criação de VIEWS

- Como criar VIEWS para guardar consultas complexas na base de dados

Criação e execução de STORED PROCEDURES

- Como criar ações na base de dados usando STORED PROCEDURES

Fundamentos de Node.js e servidores

Fundamentos do NodeJS

Instalação e configuração do NodeJS

Execução de uma aplicação no NodeJS

Diferentes tipos de módulos no NodeJS

- Core Modules
- Local Modules
- Third Party Modules

Core modules (fs, http, path, os, etc)

- Diferentes core modules incluídos no NodeJS
- Módulos como fs, para interação com o sistema de ficheiros, http para criar servidores HTTP, path para trabalhar com caminhos, etc.

Criação de aplicações web com o módulo HTTP

Ligação a bases de dados MySQL

- Ligação a uma base de dados MySQL usando o NodeJS
- Conceitos de Query Builders e ORM’s no que toca a integração de aplicações web com bases de dados

Desenvolvimento de API’s com NodeJS e Express

Conceitos de uma API REST

- Conceitos de uma API REST
- Estruturar uma api usando os recursos do protocolo HTTP, como os tipos de pedido e códigos de resposta
- Construção de endereços semânticos para uma boa API REST

Instalação do express.js

Criação de rotas

- Criação rotas no express.js
- Diferentes tipos de rota e como agrupar rotas
- Interpretação de endereços com segmentos dinâmicos

Interpretação do pedido e da resposta HTTP

- Interpretação do pedido e da resposta HTTP
- Obter os dados enviados do cliente em JSON através do pedido HTTP, tal como responder de novo para o cliente usando a resposta HTTP, também em JSON usando o código de resposta mais adequado

Criação de middlewares

- Uso dos middlewares do express e criar os próprios middlewares
- Uso de middlewares para a organização da aplicação

Autenticação com base em tokens

- Como funciona a autenticação de uma API REST, com base em tokens
- Como funciona o envio e o processamento de um token
- Diferença entre tokens assinados e tokens persistidos
- O que é um JWT e como utilizar em conjunto com o express.js

Workshop de testes automatizados com Jest

Conceitos de testes aplicacionais

Testes unitários

- Como dividir unidades de código para que possam ser testadas individualmente

Testes de integração

- Como testar a integração de várias funcionalidades em conjunto

Testes e2e

- Testes de todo um fluxo de uma aplicação
- Divisão dos use cases da aplicação para que possam ser testados de forma autónoma

Instalação e configuração do Jest

- Instalação do Jest e a configurá-lo juntamente com um projeto Javascript, para que se possa criar os testes de seguida

Assertions

- Utilidade das assertions para que possa garantir que a funcionalidade testada funciona corretamente

Isolamento de testes

- Boas práticas de testes, isolando o ambiente de execução dos testes para que não haja contaminação

Mocks e stubs

- Conceitos conceitos de mocks e stubs, para funcionalidades que necessitam de dados simulados

Cobertura de código

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

Pitstop

Projeto: API completa com integração em bases dados

Autenticação e Segurança em Aplicações Web com Node.js

Introdução à Segurança em Aplicações Web

- Visão geral das vulnerabilidades web mais comuns (OWASP Top 10)
- Importância da segurança em aplicações web

Autenticação vs. Autorização

- Diferenças chave e casos de uso de cada uma
- Exemplos em aplicações reais

Implementação de Autenticação com Node.js

- Sessões tradicionais vs. autenticação sem estado
- Estratégias de gestão de sessões e configuração segura de cookies
- Configuração de JSON Web Tokens (JWT) numa aplicação Node.js

Gestão e Segurança de Passwords

- Armazenamento seguro de passwords com encriptação
- Utilização de salting e hashing para passwords
- Fluxos de recuperação e gestão de passwords

Proteção de Dados com Encriptação

- Introdução à criptografia em Node.js
- Uso de bibliotecas para encriptação de dados

Prevenção de Vulnerabilidades Comuns de Segurança

- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- Injeção SQL
- Content Security Policy (CSP) e cabeçalhos HTTP seguros

Deployment e DevOps

Introdução ao Deployment

- Conceitos e terminologia
- Visão geral dos ambientes de deployment: desenvolvimento, staging e produção

Opções de Alojamento e Deployment

- Comparação entre fornecedores de cloud (AWS, Azure, GCP, Render, DigitalOcean, Netlify)
- Estratégias de deployment

Integração Contínua/Deployment Contínuo (CI/CD)

- Workflows de CI/CD e sua importância no desenvolvimento moderno
- Uso de ferramentas como GitHub Actions ou Jenkins para automação de testes e deployment

Containerização com Docker

- Introdução ao Docker e à containerização
- Construção e deployment de containers Docker para aplicações Node.js
- Gestão de aplicações multi-container com Docker Compose

Deployment para Plataformas Cloud

- Configuração e deployment de aplicações para plataformas de cloud
- Configuração de variáveis de ambiente e gestão de segredos da aplicação

Monitorização e Logging

- Configuração de logging básico com ferramentas como Winston
- Monitorização da saúde da aplicação

Boas Práticas de DevOps

- Automação de tarefas repetitivas
- Estratégias de backup e migração de bases de dados

Sessão de Encerramento