
Desenvolvedor Full Stack JavaScript Mean

Objetivo:

O JavaScript está em todos os lugares hoje em dia. Por isso, surgiu a ideia de criar sistemas baseados apenas em JavaScript. Assim, apenas com o conhecimento em uma linguagem de programação, o programador pode trabalhar no Front End, Back End e Banco de Dados.

MEAN é uma palavra formada pelas iniciais de cada uma das ferramentas usadas para se criar esses sistemas inteiramente em JavaScript- MongoDB (banco de dados), Express (back end), Angular (front end) e Node.js.

Neste curso teremos uma breve introdução a cada uma das ferramentas usadas, iremos aprender a integrá-las e, finalmente, subir um projeto para um servidor.

Pré-requisito:

Para o seu melhor aproveitamento do conteúdo do curso é imprescindível que você tenha conhecimento de Lógica de Programação.

Carga horária: 136 horas

Conteúdo Programático

JavaScript

- Sintaxe da linguagem
- Definição de objetos, métodos e atributos
- Tipos
- Arrays
- Classes e Objetos
- Números
- Strings
- Datas
- Expressões Regulares
- Objetos como Maps
- Funções e programação funcional
- Closures
- Tratamento de exceções
- Callbacks
- Promises
- async e await
- Uso de this
- Melhores práticas
- ECMAScript
- JSON

TypeScript

- Por que usar TypeScript?
- Classes
- Interfaces
- Tipos
- Asserção de tipos
- Inferência de tipo
- Escopo de variáveis
- Módulos
- Construtores
- Atributos e Métodos
- Arrays
- Tuples
- Unions
- Namespaces
- Módulos
- Ambientes
- Transpilação para JavaScript

NodeJS

- Configuração do ambiente
- REPL
- NPM
- package.json
- Callbacks
- Eventos

- Buffers
- Streams
- Sistema de arquivos
- require, exports e module.exports
- Objetos globais
- Módulos utilitários
- Escrevendo seus próprios módulos
- Web e Express
- Servindo conteúdo estático
- Web Services Restful com banco MongoDB
- Debug
- Segurança
- Usar TypeScript com NodeJS

MongoDB

- Introdução a bancos NoSQL
- JSON e BSON
- Vantagens
- Qual o problema com os bancos relacionais?
- Instalação e configuração do ambiente
- Coleções e Documentos
- Tipos de dados
- Modelagem de dados
- Inserção, Edição, Consulta e Exclusão de documentos em coleções
- Índices
- Replicações

- Sharding
- Relacionamentos (referenciados e embutidos)
- Agregação
- Projeções
- Escrevendo Scripts
- Administração do servidor

Angular JS

- Arquitetura de Aplicações com Angular 1
- Módulos
- Controllers
- Recipes
- Factories
- Services
- Constants
- Values
- Consumo de dados via Web Service REST/JSON
- Tabelas
- Formulários
- Expressões
- Diretivas
- Componentes
- Rotas nativas e UI Router
- Filtros
- Validação de dados
- Providers

Angular 9

- Arquitetura
- Módulos
- Componentes
- Navegação entre componentes
- Serviços
- Consumo de dados via web service REST/JSON
- Diretivas
- Templates
- User Input
- Forms
- Metadados
- Data Binding
- Event Binding
- Diretivas
- Pipes
- Injeção de Dependência
- Rotas
- CLI