



# FULLSTACK ANGULAR E SPRING

Turma de Outubro/2018



Ainda tem dúvidas?  
Fale com a gente pelo WhatsApp.

**+55 (34) 9 9861-0006**

**Clique Aqui!**

CURSO ONLINE

# FULLSTACK ANGULAR E SPRING

## 1. Introdução ao REST

- 1.1. Introdução ao curso
- 1.2. Como usar o suporte
- 1.3. O que é SOFEA?
- 1.4. O que é REST?
- 1.5. Conhecendo o projeto do curso
- 1.6. Ambiente de desenvolvimento REST
- 1.7. Testando APIs com Postman
- 1.8. Introdução ao protocolo HTTP

## 2. Fundamentos do REST

- 2.1. O que é um recurso?
- 2.2. Representações de um recurso
- 2.3. Modelo de maturidade Richardson - Nível 0
- 2.4. Modelo de maturidade Richardson - Nível 1
- 2.5. Modelo de maturidade Richardson - Nível 2
- 2.6. Modelo de maturidade Richardson - Nível 3
- 2.7. HATEOAS
- 2.8. Segurança de APIs REST
- 2.9. Idempotência

## 3. Primeiras consultas e cadastros na API

- 3.1. Criando o projeto da API
- 3.2. Conectando ao MySQL
- 3.3. Migração de dados com Flyway
- 3.4. Consultando primeiro recurso com GET
- 3.5. Coleção vazia, o que retornar?
- 3.6. Cadastrando nova categoria com POST
- 3.7. Desafio: Retornar 404 caso não exista a categoria
- 3.8. Validando atributos desconhecidos
- 3.9. Tratando erros com ExceptionHandler
- 3.10. Validando valores inválidos com Bean Validation
- 3.11. Desafio: Criando o cadastro de pessoa
- 3.12. Usando eventos para adicionar header Location

## 4. Atualização e remoção de recursos na API

- 4.1. Removendo pessoa com DELETE
- 4.2. Sobre atualização de recursos REST
- 4.3. Atualizando pessoa com PUT
- 4.4. Implementando atualização parcial com PUT

## 5. Relacionamentos entre recursos REST

- 5.1. Criando a migração e entidade de lançamento

- 5.2. Desafio: Lista e busca de lançamentos
- 5.3. Desafio: Cadastrando o primeiro lançamento
- 5.4. Validando inconsistências
- 5.5. Validando lançamento com Bean Validation
- 5.6. Regra para não salvar pessoa inativa
- 5.7. Implementando pesquisa de lançamento com Metamodel
- 5.8. Desafio: Removendo lançamentos
- 5.9. Implementando a paginação de lançamentos

## **6. Segurança da API**

- 6.1. Implementando autenticação Basic
- 6.2. Fluxo básico do OAuth
- 6.3. Implementando segurança com OAuth 2 e Password Flow
- 6.4. JSON Web Tokens - JWT
- 6.5. Configurando JWT no projeto
- 6.6. Renovando o access token com o refresh token
- 6.7. Movendo o refresh token para o cookie
- 6.8. Movendo o refresh token do cookie para a requisição
- 6.9. O que é CORS?
- 6.10. Criando filtro para CORS
- 6.11. Movendo o usuário para o banco de dados
- 6.12. Adicionando permissões de acesso
- 6.13. Desafio: Finalizando permissões de acesso
- 6.14. Implementando o logout

## **7. Deploy da API em produção**

- 7.1. Implementando projeção de lançamento
- 7.2. Profiles do Spring
- 7.3. Criando a conta no Heroku
- 7.4. Deploy da API na nuvem
- 7.5. Nome do usuário no token JWT
- 7.6. Alternando OAuth 2 e Basic Security com profiles
- 7.7. Desafio: Pesquisa de pessoa
- 7.8. Ajustando o CEP
- 7.9. Desafio: Atualização de lançamento

## **8. Introdução ao Angular**

- 8.1. O que é Angular?
- 8.2. AngularJS vs Angular 2/4/X: a confusão das versões
- 8.3. Instalando o Visual Studio Code
- 8.4. Introdução ao HTML
- 8.5. Introdução ao CSS
- 8.6. Instalando o Node.js e NPM
- 8.7. Instalando e criando um projeto com Angular CLI
- 8.8. Abrindo o projeto no VS Code
- 8.9. Abrindo e executando um exemplo do curso

## **9. Fundamentos do Angular, componentes e data binding**

- 9.1. Bootstrapping e AppModule
- 9.2. O que são componentes
- 9.3. Criando um componente
- 9.4. Instalando a biblioteca CSS do Bootstrap
- 9.5. Introdução a data binding
- 9.6. Usando interpolação
- 9.7. Usando event binding
- 9.8. Usando variável de referência
- 9.9. Usando property binding
- 9.10. Usando two-way data binding
- 9.11. Introdução às diretivas
- 9.12. Exibindo condicionalmente com as diretivas ngIf e hidden
- 9.13. Iterando com a diretiva ngFor
- 9.14. Binding de propriedades customizadas com @Input
- 9.15. Binding de eventos customizados com @Output e EventEmitter
- 9.16. Adicionando estilos CSS em componentes
- 9.17. Estilos CSS dinâmicos com ngStyle
- 9.18. Classes CSS dinâmicas com ngClass

## **10. Páginas de pesquisa**

- 10.1. Instalando plugins úteis no Visual Studio Code
- 10.2. Escolhendo uma biblioteca de componentes
- 10.3. Criando o projeto do curso e instalando o PrimeNG
- 10.4. Adicionando o formulário de pesquisa de lançamentos
- 10.5. Adicionando uma tabela de dados
- 10.6. Customizando colunas com ng-template
- 10.7. Fazendo paginação de dados
- 10.8. Adicionando tooltip
- 10.9. Colocando a tabela de dados responsiva
- 10.10. Criando o componente de pesquisa de lançamentos
- 10.11. Criando o componente de barra de navegação
- 10.12. Adicionando menu intercambiável
- 10.13. Desafio: criando componente de pesquisa de pessoas

## **11. Diretivas e pipes**

- 11.1. Criando diretivas customizadas
- 11.2. Respondendo a eventos do hospedeiro com @HostListener
- 11.3. Vinculando propriedades do hospedeiro com @HostBinding
- 11.4. Usando property binding em diretivas customizadas
- 11.5. Exportando a API da diretiva para o template
- 11.6. Conhecendo e usando pipes
- 11.7. Passando parâmetros para pipes
- 11.8. Desafio: usando pipes

## **12. Formulários e validação**

- 12.1. Introdução aos formulários
- 12.2. Template-driven Forms: Criando um formulário
- 12.3. Registrando os controles do formulário

- 12.4. Adicionando opções dinâmicas no campo de seleção
- 12.5. Definindo o valor padrão em campos com ngModel
- 12.6. Two-way binding com ngModel
- 12.7. Adicionando validação em formulários
- 12.8. Exibindo erro de validação do formulário
- 12.9. Exibindo erro de validação de controles do formulário
- 12.10. Rastreamento do estado em controles do formulário
- 12.11. Estilizando os campos inválidos com classes CSS do Angular
- 12.12. Estilizando os campos inválidos com Bootstrap
- 12.13. Limpando formulários (reset)

### **13. Páginas de cadastro**

- 13.1. Criando o protótipo do formulário de cadastro de lançamentos
- 13.2. Adicionando seletor de data (componente Calendar)
- 13.3. Adicionando botão de seleção
- 13.4. Adicionando caixa de seleção (componente Dropdown)
- 13.5. Adicionando máscara de dinheiro com ng2-mask-money
- 13.6. Desafio: criando o protótipo do formulário de cadastro de pessoa
- 13.7. Adicionando campo com máscara (componente InputMask)
- 13.8. Validando controles de formulário com PrimeNG
- 13.9. Criando componente de mensagem de erro de validação
- 13.10. Desafio: controles, validações e mensagens de erro
- 13.11. Desafio: criando mais componentes

### **14. Módulos do Angular**

- 14.1. Introdução aos módulos
- 14.2. Criando um módulo e exportando um componente
- 14.3. Reexportando um módulo
- 14.4. Criando um componente interno do módulo
- 14.5. O que são Feature Modules?
- 14.6. Criando um Feature Module
- 14.7. Desafio: criando o feature module de pessoas
- 14.8. O que são Shared Modules?
- 14.9. Criando um Shared Module
- 14.10. O que é Core Module?
- 14.11. Desafio: criando o Core Module

### **15. Serviços e injeção de dependências**

- 15.1. Introdução aos serviços
- 15.2. Implementando um serviço
- 15.3. O que é injeção de dependências?
- 15.4. Configurando o injetor com provider por classe
- 15.5. Configurando o injetor com provider por fábrica
- 15.6. Configurando o injetor com provider por valor e o decorator @Inject
- 15.7. Injetando serviços dentro de serviços e o decorador @Injectable
- 15.8. Como funciona o Injetor Hierárquico

### **16. Requisições HTTP**

- 16.1. Por que precisamos de requisições HTTP?
- 16.2. Instalando e testando o json-server
- 16.3. Fazendo requisição com GET e recebendo o retorno
- 16.4. Fazendo requisição com POST
- 16.5. Fazendo requisição com DELETE
- 16.6. Fazendo requisição com PUT
- 16.7. Tratando erros de requisições HTTP

## **17. Implementando os serviços do projeto**

- 17.1. Revisando e iniciando o back-end do projeto do curso
- 17.2. Criando o serviço de consulta de lançamentos
- 17.3. Adicionando filtro por descrição na pesquisa de lançamentos
- 17.4. Adicionando filtro por datas na pesquisa de lançamentos
- 17.5. Implementando a paginação no serviço de lançamentos
- 17.6. Configurando a paginação lazy do PrimeNG
- 17.7. Desafio: criando a consulta e listagem de pessoas
- 17.8. Excluindo lançamentos e o decorador @ViewChild
- 17.9. Adicionando mensagem de sucesso com Angular Toasty
- 17.10. Adicionando diálogo de confirmação antes da exclusão
- 17.11. Alterando o locale da aplicação para pt-BR
- 17.12. Criando um serviço de tratamento de erros
- 17.13. Desafio: implementando a exclusão de pessoas
- 17.14. Desafio: mensagem de erro de usuário na exclusão de pessoa
- 17.15. Desafio: implementando a mudança de status de pessoas
- 17.16. Desafio: implementando o serviço de listagem de categorias
- 17.17. Listando as categorias cadastradas no dropdown
- 17.18. Desafio: listando as pessoas cadastradas no dropdown
- 17.19. Criando classes de modelo e usando no cadastro de lançamentos
- 17.20. Implementando o serviço de cadastro de lançamentos
- 17.21. Desafio: implementando o cadastro de pessoas

## **18. Roteamento e navegação**

- 18.1. Introdução a rotas
- 18.2. Configurando rotas na aplicação
- 18.3. Navegando com Router Link
- 18.4. Estilizando links da rota ativa
- 18.5. Recebendo parâmetros da rota
- 18.6. Desafio: implementando os serviços de atualização e busca por código
- 18.7. Preenchendo os campos na edição de lançamentos
- 18.8. Salvando lançamentos editados
- 18.9. Implementando navegação imperativa
- 18.10. Fazendo redirecionamento
- 18.11. Tratando rota não encontrada
- 18.12. Definindo o título da página dinamicamente
- 18.13. Refatorando as rotas para usar Routing Module
- 18.14. Criando um Routing Module para o módulo de funcionalidade
- 18.15. Desafio: roteamento e edição de pessoas

## **19. Segurança do front-end**

- 19.1. Introdução à segurança do front-end
- 19.2. Revisando a segurança da API com OAuth 2 e JWT
- 19.3. Desafio: módulo de segurança e protótipo da tela de login
- 19.4. Implementando o serviço de autenticação com OAuth 2
- 19.5. Decodificando o JWT e armazenando no Local Storage
- 19.6. Tratando casos de erros e sucesso de autenticação
- 19.7. Adicionando o Access Token nas chamadas HTTP
- 19.8. Exibindo o nome do usuário logado
- 19.9. Exibindo o menu do sistema conforme permissões do usuário
- 19.10. Obtendo um novo access token
- 19.11. Interceptando chamadas HTTP para tratar a expiração do access token
- 19.12. Protegendo componentes
- 19.13. Protegendo rotas com guarda de rotas (CanActivate)
- 19.14. E se o Refresh Token expirar?
- 19.15. Tratando acessos de usuários deslogados na AuthGuard
- 19.16. Implementando o logout

## **20. Deploy em produção do cliente Angular**

- 20.1. Configurando a aplicação com environment do Angular CLI
- 20.2. Fazendo build para o ambiente de produção
- 20.3. Respondendo requisições com Node.js e Express
- 20.4. Fazendo deploy em produção no Heroku
- 20.5. Conclusão

## **21. Apêndice: Atualizações e recursos avançados**

- 21.1. Atualizando para o Angular 5
- 21.2. Corrigindo problemas com Locale
- 21.3. Atualizando o PrimeNG
- 21.4. Alterando de DataTable para TurboTable
- 21.5. Exercício: utilizando TurboTable na pesquisa de pessoas
- 21.6. Carregamento tardio de módulos (Lazy loading)
- 21.7. Formulários reativos
- 21.8. Criando um formulário reativo
- 21.9. Usando a propriedade formGroup
- 21.10. Configurando o HTML do formulário reativo
- 21.11. Criando validações customizadas

## **22. Apêndice: Melhorando o back-end**

- 22.1. Preparação do retorno dos dados para os gráficos
- 22.2. Criando consulta para dados por categoria
- 22.3. Retornando os dados estatísticos de lançamento por categoria
- 22.4. Criando consulta para dados por dia
- 22.5. Retornando os dados estatísticos de lançamento por dia
- 22.6. Instalando o Jaspersoft Studio
- 22.7. Ajustando o layout do relatório
- 22.8. Criando o DTO do relatório
- 22.9. Criando os campos e parâmetros do relatório

- 22.10. Ajustando o título e o rodapé do relatório
- 22.11. Usando os campos do DTO no relatório
- 22.12. Criando a consulta do relatório
- 22.13. Gerando os bytes do relatório
- 22.14. Retornando os bytes do relatório na requisição
- 22.15. Criando um agendamento de tarefa (Scheduler)
- 22.16. Configurando o envio de e-mail
- 22.17. Enviando um e-mail simples
- 22.18. Configurando o template para o envio do e-mail
- 22.19. Processando o template e enviando o e-mail
- 22.20. Buscando lançamentos vencidos com Spring Data JPA
- 22.21. Agendando o envio de e-mail
- 22.22. Incluindo logs no agendamento do envio de e-mail
- 22.23. Criando a entidade Contato para suportar mestre-detalle
- 22.24. Resolvendo o StackOverflowError com @JsonIgnoreProperties
- 22.25. Inserindo uma pessoa com contato
- 22.26. Usando a propriedade orphanRemoval
- 22.27. Ignorando contatos da pessoa na pesquisa de lançamento
- 22.28. Upload de arquivos para API
- 22.29. Criando conta na Amazon AWS
- 22.30. Configurando o serviço S3
- 22.31. Criando o bucket no S3 automaticamente
- 22.32. Implementando o envio do arquivo para o S3
- 22.33. Enviando arquivos para o S3
- 22.34. Anexando arquivo no lançamento
- 22.35. Atualizando e removendo anexo
- 22.36. Configurando URL do anexo

### **23. Apêndice: Melhorando o front-end**

- 23.1. Criando o módulo Dashboard
- 23.2. Plotando gráficos com dados estáticos
- 23.3. Criando o serviço da Dashboard
- 23.4. Buscando dados do gráfico de pizza
- 23.5. Buscando dados do gráfico de linhas
- 23.6. Formatando labels no Chart.JS
- 23.7. Criando módulo de relatórios
- 23.8. Configurando formulário do relatório
- 23.9. Exibindo o PDF para o usuário
- 23.10. Exercício: Incluindo itens de menu: dashboard e lançamento
- 23.11. Listando contatos na tela mestre-detalle
- 23.12. Criando o diálogo de contato
- 23.13. Criando o formulário de contato
- 23.14. Incluindo um novo contato
- 23.15. Corrigindo estilo do botão "Novo"
- 23.16. Editando contato
- 23.17. Removendo contato
- 23.18. Criando componente de contatos
- 23.19. Upload com o componente FileUpload



- 23.20. Fazendo download do anexo
- 23.21. Tratando erro de upload
- 23.22. Utilizando componente ProcessSpinner
- 23.23. Desabilitando botão "Salvar" no upload
- 23.24. Salvando e removendo anexo

## **24. Apêndice: Combos dependentes**

- 24.1. Criando entidades cidade e estado
- 24.2. Criando pesquisa de estados e cidades
- 24.3. Buscando estados e cidades
- 24.4. Preenchendo Dropdown de estados
- 24.5. Carregando Dropdown de cidades
- 24.6. Validando cidade e estado, e salvando pessoa
- 24.7. Ajustando estado e cidade na pesquisa de pessoas

## **25. Apêndice: Angular 6 e Spring Boot 2**

- 25.1. Atualizando para Spring Boot 2
- 25.2. Novas assinaturas do Spring Data JPA
- 25.3. Modificações para o Spring Security 5
- 25.4. Atualizando para o Angular 6
- 25.5. Corrigindo ícones
- 25.6. Usando o Growl
- 25.7. Atualizando o módulo JWT
- 25.8. Alterando a classe MoneyHttp
- 25.9. Usando a classe MoneyHttp
- 25.10. Corrigindo o link de mudança de status
- 25.11. Rodando comando ng build

---

CURSO ONLINE

# **WEB DESIGN RESPONSIVO COM HTML5, CSS3 E BEM (BÔNUS)**

## **1. Introdução**

- 1.1. Introdução ao curso
- 1.2. Apresentando o projeto
- 1.3. Preparando o ambiente de desenvolvimento

## **2. Começando com HTML**

- 2.1. O que é HTML?
- 2.2. Estrutura básica do documento
- 2.3. DOCTYPE e codificação

- 2.4. Primeiras tags: títulos, quebras de linhas e parágrafos
- 2.5. Comentários
- 2.6. Ênfase, importância e marcação
- 2.7. Imagens
- 2.8. Âncoras (links)
- 2.9. Elementos estruturais

### **3. Começando com CSS**

- 3.1. O que é CSS
- 3.2. Estilos incorporados e Regras CSS
- 3.3. Estilos em arquivos externos
- 3.4. Seletores de tipo, classe e ID
- 3.5. Agrupando seletores
- 3.6. Seletores descendentes
- 3.7. Seletores de filhos diretos
- 3.8. Cores
- 3.9. Formatação de textos
- 3.10. Inspeccionando com Chrome DevTools
- 3.11. Entendendo a propriedade display
- 3.12. Adicionando bordas
- 3.13. Espaçamento interno (padding)
- 3.14. Margens de elementos
- 3.15. Box model e a propriedade box-sizing

### **4. Iniciando o projeto do curso**

- 4.1. Preparando o projeto
- 4.2. Criando o cabeçalho da página
- 4.3. Listas ordenadas e não-ordenadas
- 4.4. Adicionando os planos
- 4.5. Flutuando elementos
- 4.6. Usando pseudo-elementos
- 4.7. Configurando os planos lado a lado com float
- 4.8. Criando um botão
- 4.9. As pseudo-classes :focus e :hover
- 4.10. Reset CSS e Normalize.css

### **5. Web Design Responsivo**

- 5.1. O que é Responsive Web Design?
- 5.2. Unidade de medida: pixel
- 5.3. Unidade de medida: percentual
- 5.4. Meta tag viewport
- 5.5. Layout fixo e fluído
- 5.6. Layout responsivo e media queries
- 5.7. Como funciona um sistema de Grid CSS
- 5.8. Sistema de grid do Bootstrap
- 5.9. Ajustando o projeto para usar Grid CSS

### **6. Especificidade, BEM e boas práticas**

- 6.1. Especificidade do CSS
- 6.2. Caos no CSS: porque uma metodologia é importante?
- 6.3. A Metodologia BEM: seu código escalável
- 6.4. Como usar BEM na prática - parte 1
- 6.5. Como usar BEM na prática - parte 2
- 6.6. Ajustando o projeto do curso com BEM
- 6.7. Mais organização: CSS com Guidelines

## **7. Encerrando o projeto e mais CSS**

- 7.1. Unidades de medida: em e rem
  - 7.2. Ajustando unidades de medida no projeto
  - 7.3. Adicionando chamada principal
  - 7.4. Adicionando depoimento
  - 7.5. Adicionando rodapé
  - 7.6. Posicionamento estático e fixo
  - 7.7. Posicionamento relativo
  - 7.8. Posicionamento absoluto
  - 7.9. Adicionando rótulo no plano
  - 7.10. Ajustando margem do plano
  - 7.11. Adicionando aspas no depoimento
  - 7.12. Adicionando o bloco de navegação
  - 7.13. Adicionando o menu para telas pequenas
  - 7.14. Ajustando o menu para telas médias e grandes
  - 7.15. JavaScript Hook: chaveando o menu
  - 7.16. Criando um formulário: assinatura de plano
  - 7.17. Concluindo o formulário de assinatura de plano
  - 7.18. Entendendo as tabelas do HTML
  - 7.19. Aplicando estilos em tabelas com CSS
  - 7.20. Conclusão e próximos passos
- 

CURSO ONLINE

# **COMEÇANDO COM JAVASCRIPT (BÔNUS)**

## **1. Introdução**

- 1.1. Introdução ao JavaScript
- 1.2. Introdução ao workshop
- 1.3. Preparando o ambiente

## **2. Conceitos básicos**

- 2.1. Sintaxe básica
- 2.2. Tipos de dados
- 2.3. Condicionais
- 2.4. Loops

## 2.5. Hoisting

### 3. Funções

- 3.1. Declaração
- 3.2. Invocação
- 3.3. Auto invocação
- 3.4. Closures

### 4. Objetos

- 4.1. Definição
- 4.2. Prototype
- 4.3. Arrays
- 4.4. Date

### 5. Funções úteis

- 5.1. Manipulando Strings
- 5.2. Manipulando números
- 5.3. Manipulando arrays

### 6. Conclusão

- 6.1. Boas práticas
  - 6.2. Conclusão
- 

## CURSO ONLINE

# PROGRAMANDO EM TYPESCRIPT (BÔNUS)

### 1. Introdução

- 1.1. Introdução ao curso
- 1.2. O que é TypeScript?
- 1.3. Instalando o Node.js e NPM
- 1.4. Conhecendo e instalando o Visual Studio Code
- 1.5. Instalando o TypeScript
- 1.6. Primeiro código com TypeScript

### 2. Fundamentos do TypeScript

- 2.1. Declaração de variáveis (var, let e const)
- 2.2. Tipos básicos
- 2.3. Inferência de tipos
- 2.4. Arrays e iterações

### 3. Orientação a objetos

- 3.1. Classes e objetos
- 3.2. Propriedades e métodos

- 3.3. Construindo sua primeira classe
- 3.4. Herança
- 3.5. Interfaces

#### **4. Outros recursos do ES6**

- 4.1. Fat arrow functions
- 4.2. Template strings
- 4.3. Promise
- 4.4. Conclusão