

Introdução a Teste de Software com Springboot

Samuel Lucas

Roteiro

- Conhecimentos necessários
- Metodologia do minicurso
- Ferramentas instaladas
- Conclusão

Conhecimentos necessários



[Fonte](#)

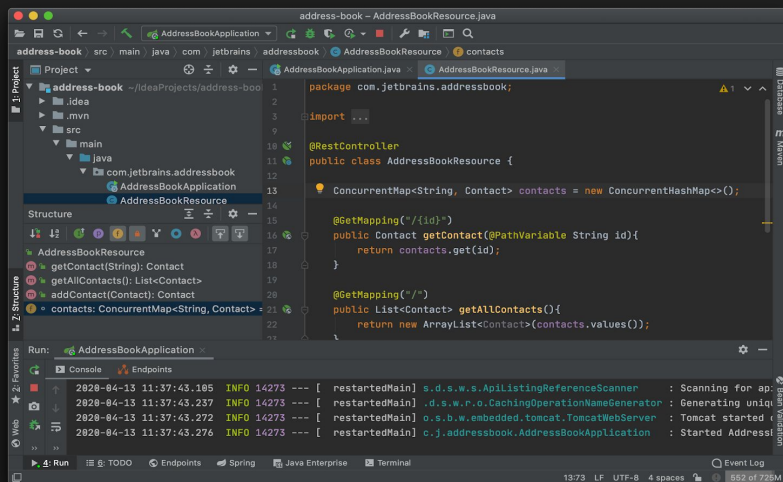
Metodologia do minicurso

- Minicurso online de 7 módulos
- Slides + Implementação de códigos
- Material disponibilizado no [Github](#)

Ferramentas necessárias

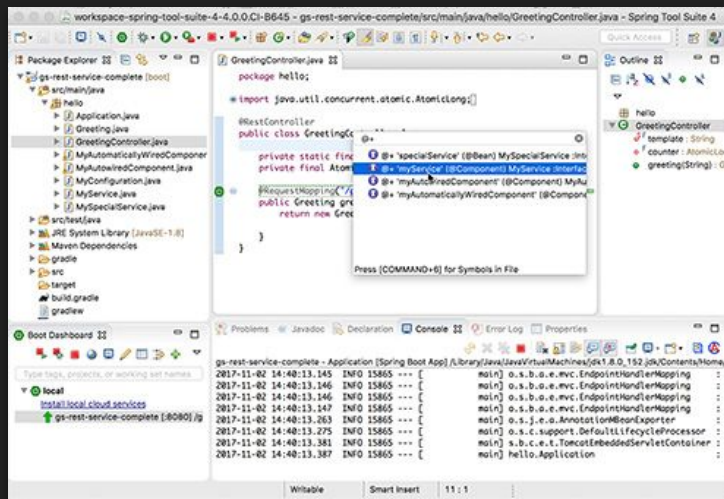
(1/3)

Fonte



IntelliJ

Fonte



Spring Tool Suite

Ferramentas necessárias

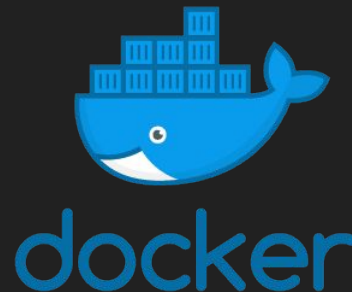
(2/3)

[Fonte](#)



+

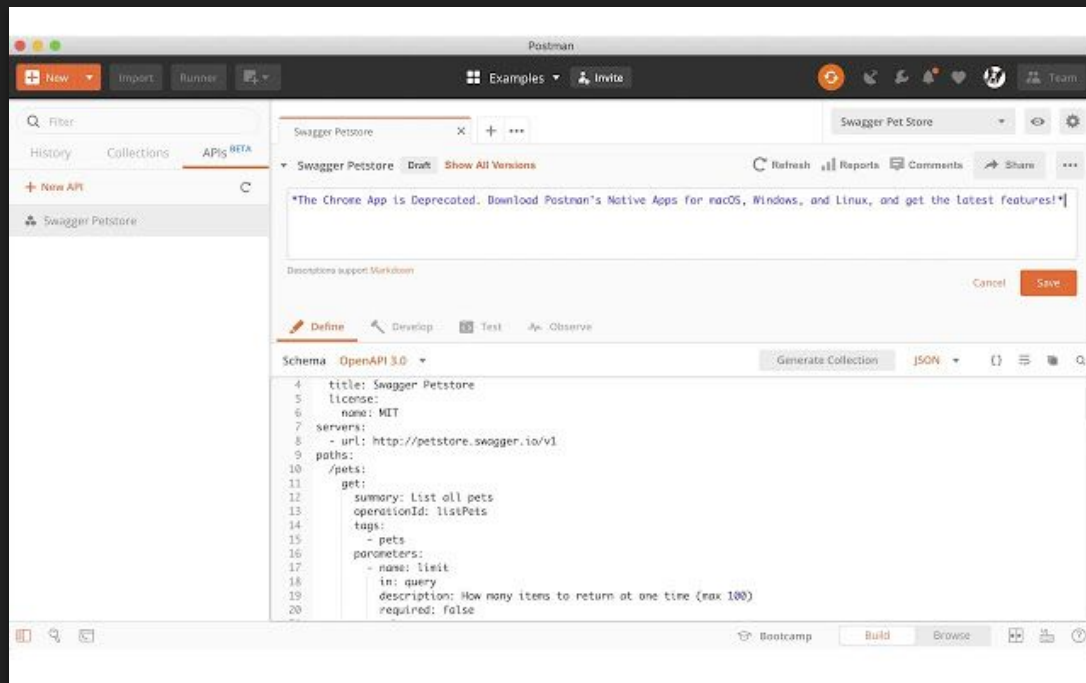
[Fonte](#)



<https://imasters.com.br/data/utilizando-docker-com-mysql>

Ferramentas necessárias

(3/3)



[Fonte](#)

Conteúdo dos módulos

(1/2)

- **Módulo 1: Introdução a teoria de Teste de Software**

- Motivação
- O que é Teste de Software?
- A adição de testes podem melhorar a qualidade de um software?
- Por que os testes não são aplicados em larga escala?
- Caso de teste
- Níveis de teste

- **Módulo 2: Framework de automação de testes**

- Testes automatizados
- JUnit
- JUnit: qual versão usaremos?
- Comando e métodos JUnit
- JUnit 4 x JUnit 5

- **Módulo 3: Configurando dependências de teste**

- Spring Boot Start Test
- Banco de dados em memória
- Passeando por projeto Springboot

Conteúdo dos módulos

(2/2)

- **Módulo 4: Início da codificação dos testes unitários da aplicação**
 - Aplicação: Sistema de notas
 - Teste manuais
 - Passos de teste
 - Codificação dos teste unitários da camada Repository
- **Módulo 5: Continuação da codificação dos testes unitários da aplicação**
 - Ambiente de teste
 - Dublês de teste
 - Mockito
 - Codificação dos teste unitários da camada Service
- **Módulo 6: Finalização da codificação dos testes unitários da aplicação**
 - Codificação dos teste unitários da camada Controller
 - Teste funcional
 - Teste estrutural
 - JaCoco
 - Teste baseado em defeitos
 - Teste de regressão
- **Módulo 7: Teste de integração**
 - Codificação dos teste da camada Controller
 - Testes parametrizados