

Setup inicial do projeto

Seja bem vindo a última parte do curso! Esse primeiro exercício é extenso e não necessariamente fácil. Seja paciente, vá com calma e confira cada passo que você der.

Instalação do Java

Precisamos ter instalado o Java 8 na nossa máquina. Minha sugestão é que você um projeto chamado **SDKMAN**. Para baixar e instalar você pode acessar o endereço <https://sdkman.io/install> (<https://sdkman.io/install>). Lá existe a explicação para os 3 sistemas operacionais mais famosos. O segundo passo é instalar o Java e você pode encontrar informações nesse outro endereço <https://sdkman.io/usage> (<https://sdkman.io/usage>).

Lembre, é **obrigatório** o uso do Java 8. Se você tiver instalado o Java 9 ou 10, encontrará problemas relativas a parte de módulos da linguagem.

Caso não queira usar a sugestão acima, também pode acessar o site da oracle e baixar a versão relativa a seu sistema operacional. O endereço é <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>).

Para te ajudar um pouco a entender sobre o básico sobre Java e processo de instalação, acesse a segunda aula do curso de introdução a Java. O endereço é <https://cursos.alura.com.br/course/java-primeiros-passos/task/34514> (<https://cursos.alura.com.br/course/java-primeiros-passos/task/34514>).

Depois de instalado, você deve rodar o comando `java -version` no seu terminal e a saída deve ter o Java 8.

Instalação do MySQL

Para usuários windows, sugerimos que acesse a aula <https://cursos.alura.com.br/course/introducao-a-banco-de-dados-e-sql/task/5652> (<https://cursos.alura.com.br/course/introducao-a-banco-de-dados-e-sql/task/5652>). Caso você use mac ou ubuntu, acesse a aula <https://cursos.alura.com.br/course/introducao-a-banco-de-dados-e-sql/task/5653> (<https://cursos.alura.com.br/course/introducao-a-banco-de-dados-e-sql/task/5653>).

Executando a aplicação Java

Primeiro precisamos fazer o download do arquivo referente a aplicação java em si. Ele pode ser baixado através do endereço <https://github.com/alberto-alura/curso-introducao-blockchain/raw/master/eleicoesonline.war> (<https://github.com/alberto-alura/curso-introducao-blockchain/raw/master/eleicoesonline.war>).

Também precisamos baixar o arquivo necessário para executar o servidor web java. O download pode ser feito através do endereço <https://github.com/alberto-alura/curso-introducao-blockchain/raw/master/webapp-runner.jar> (<https://github.com/alberto-alura/curso-introducao-blockchain/raw/master/webapp-runner.jar>).

Uma vez baixados, deixe os dois na mesma pasta. Agora, você precisa abrir seu terminal, navegar até a pasta onde se encontram os dois arquivos e executar o comando abaixo:

```
java -Dspring.datasource.username=root -Dspring.datasource.password= -jar webapp-runner.jar --port
```

Nesse comando é feita a suposição que seu mysql foi instalado com o usuário **root** e com a senha vazia. Se tiver uma senha de acesso, é só passar como parâmetro. Ex:

```
java -Dspring.datasource.username=root -Dspring.datasource.password=SUA SENHA AQUI -jar webapp-runner
```

Se tudo deu certo, você vai uma última mensagem no log parecida com a que segue.

```
Started Boot in 16.208 seconds
```

Gerando os dados iniciais

Se tudo deu certo até aqui, você já pode acessar alguns endereços referentes a aplicação Java. Nesse momento, vamos acessar dois endereços para a geração dos dados mínimos para a utilização.

O primeiro deve ser <http://localhost:8080/magic/generate/roles> (<http://localhost:8080/magic/generate/roles>). O segundo vai ser <http://localhost:8080/magic/generate/owner/> (<http://localhost:8080/magic/generate/owner/>).

O primeiro endereço gera os perfis de acesso da nossa aplicação. O segundo gera um usuário capaz de criar eleições, aprovar candidatos etc.

Instalação do Node

Caso tenha gostado do **SDKMAN**, você pode usá-lo para instalar o Node também. Lembre que o curso foi gravado com a versão 8, então a sugestão é que você instale a mesma.

Uma outra opção é acessar o endereço <https://nodejs.org/en/download/> (<https://nodejs.org/en/download/>) e fazer o download da versão 8 diretamente por lá.

Instalação do Docker

Para instalar o Docker você pode usar a própria explicação da Alura. Para o windows, acesse o endereço <https://cursos.alura.com.br/course/docker-e-docker-compose/task/29235> (<https://cursos.alura.com.br/course/docker-e-docker-compose/task/29235>).

Para instalar no macOS, você pode acessar o endereço <https://cursos.alura.com.br/course/docker-e-docker-compose/task/29235> (<https://cursos.alura.com.br/course/docker-e-docker-compose/task/29235>).

Para instalar no ubuntu, você pode acessar o endereço <https://cursos.alura.com.br/course/docker-e-docker-compose/task/28593> (<https://cursos.alura.com.br/course/docker-e-docker-compose/task/28593>).

Instalação do Docker Compose

Para a instalação do Docker Compose, você pode acessar o endereço <https://docs.docker.com/compose/install/#install-compose> (<https://docs.docker.com/compose/install/#install-compose>). Lá tem as opções para os sistemas operacionais mais famosos.

Executando o Sawtooth

Com o Docker e o Docker compose instalados, chegou a hora de subir toda infraestrutura necessária para rodar o sawtooth. Salve o conteúdo do endereço <https://raw.githubusercontent.com/alberto-alura/curso-introducao-blockchain/master/sawtooth-default.yaml> (<https://raw.githubusercontent.com/alberto-alura/curso-introducao-blockchain/master/sawtooth-default.yaml>) num arquivo chamado **sawtooth-default.yaml**. Copie exatamente da mesma forma que é apresentado na página.

Pelo terminal, navegue até a pasta onde você criou o arquivo do compose. De dentro dessa pasta, execute o seguinte comando:

```
docker-compose -f sawtooth-default.yaml up
```

Isso deve ser suficiente para subir o sawtooth. Quando quiser parar, pressione `ctrl+c` para Linux ou Mac e `ctrl+d` para Windows. Depois de parar, sempre rode o comando `docker-compose -f sawtooth-default.yaml down` para garantir que a estrutura gerada pelo docker realmente foi destruída.

Sempre que você fizer alguma alteração na aplicação Javascript, eu sugiro parar e subir a instância do docker.

Detalhes sobre o compose file

O conteúdo do arquivo executado pelo Docker compose é igual ao que segue:

```
version: "2.1"

services:

  settings-tp:
    image: hyperledger/sawtooth-settings-tp:1.0
    container_name: sawtooth-settings-tp-default
    depends_on:
      - validator
    entrypoint: settings-tp -vv -C tcp://validator:4004

  validator:
    image: hyperledger/sawtooth-validator:1.0

    container_name: sawtooth-validator-default
    expose:
      - 4004
    ports:
      - "4004:4004"
    # start the validator with an empty genesis batch
    entrypoint: "bash -c \"\
      sawadm keygen && \
      sawtooth keygen my_key && \
      sawset genesis -k /root/.sawtooth/keys/my_key.priv && \
      sawadm genesis config-genesis.batch && \
      sawtooth-validator -vv \
      --endpoint tcp://validator:8800 \
      --bind component:tcp://eth0:4004 \
      --bind network:tcp://eth0:8800 \
    \""

  rest-api:
    image: hyperledger/sawtooth-rest-api:1.0
```

```

container_name: sawtooth-rest-api-default
ports:
  - "8008:8008"
depends_on:
  - validator
entrypoint: sawtooth-rest-api -C tcp://validator:4004 --bind rest-api:8008

shell:
  image: hyperledger/sawtooth-all:1.0
  container_name: sawtooth-shell-default
  depends_on:
    - rest-api
  entrypoint: "bash -c \"\
    sawtooth keygen && \
    tail -f /dev/null \
  \""

```

A parte importante é a configuração do validator e rest-api. O primeiro vai executar nossos smart contracts dentro do Sawtooth e o segundo sobe um servidor web que já tem os endereços que podem ser acessados para manipularmos o blockchain do sawtooth. Um ponto importante é que o arquivo está configurado para sempre subir um blockchain sem nenhuma informação. A ideia é a gente testar sobre um ambiente limpo. Lembre que esse curso é uma introdução, muito mais pode ser explorado, mas vamos passar primeiro por essa fase.

Rodando a aplicação Javascript

Para executar o código da parte Javascript da aplicação, você deve começar baixando a versão inicial do projeto. O endereço https://github.com/alberto-alura/curso-introducao-blockchain/releases/tag/versao_inicial (https://github.com/alberto-alura/curso-introducao-blockchain/releases/tag/versao_inicial) pode ser acessado para esse fim.

Uma vez que você fez o download, siga os seguintes passos:

- Faça a extração dos arquivos do zip para uma pasta do seu gosto.
- Abra uma outra aba no seu terminal ou abra outro terminal mesmo.
- Navegue até a pasta de download do projeto.
- Uma vez na pasta, navegue para a pasta blockchain-api.
- Como esse é o primeiro acesso, é necessário instalar as dependências do projeto. Execute o comando `npm install .`
- Agora execute o comando `node index.js` .
- Você deve ver uma saída parecida com essa:

```
restify listening at http://[::]:8084
```