Engenharia de Telecomunicações — Prof. Emerson Ribeiro de Mello

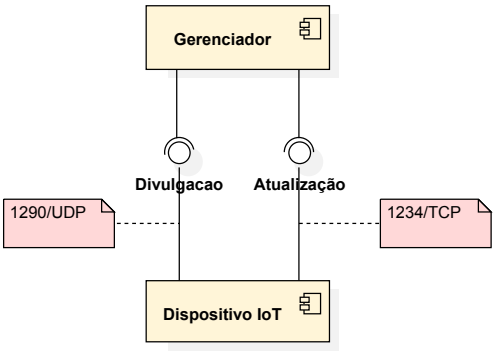
STD29006 – Sistemas Distribuídos — mello@ifsc.edu.br

Lista 1: Prática com contêineres e sockets TCP/IP em Java

23/10/2025

Desenvolva uma solução para o cenário apresentado na Figura 1, onde um dispositivo IoT publica periodicamente, em um endereço de multicast, a versão do software que está executando. Na rede local, existe um gerenciador de atualizações que monitora essas mensagens e mantém uma lista dos dispositivos IoT que estão ativos, bem como a versão do software que cada um está executando.

Figura 1: Atualização de IoT com Sockets TCP e UDP



O usuário do gerenciador de atualizações pode, a qualquer momento, solicitar: a lista de dispositivos IoT ativos e suas respectivas versões de software; incrementar a versão do software que deve ser executada pelos dispositivos IoT; e aplicar a atualização imediata de todos os dispositivos IoT para a versão mais recente do software; finalizar o gerenciador de atualizações.

O processo de atualização do software em um dispositivo IoT envolve o gerenciador de atualizações conectando na porta 1234/TCP do dispositivo IoT e enviando o novo software. O dispositivo IoT deve então substituir o software antigo pelo novo. Para este exercício, você pode simular o software como uma simples string representando a versão do software, como `v1.0`, `v1.1`, etc.

No repositório do GitHub Classroom, você deve criar dois projetos Java independentes, ambos utilizando Gradle: um para o dispositivo IoT e outro para o gerenciador de atualizações. Cada projeto deve estar em um diretório separado dentro do repositório.

```
.
|-- .gitignore
|-- docker-compose.yml
|-- Readme.md
|-- dispositivo-iot <-- aqui terá um projeto Java com gradle
`-- gerenciador-atualizacoes <-- aqui terá um projeto Java com gradle
```

Abaixo estão os requisitos que você deve atender:

1. O dispositivo IoT deve enviar periodicamente (a cada 10 segundos) uma mensagem UDP de multicast para o endereço 231.0.0.1:1290, contendo a versão do software que está executando;
2. O gerenciador de atualizações deve escutar o endereço UDP de multicast 231.0.0.1:1290 e processar as mensagens recebidas, atualizando a lista de dispositivos IoT ativos e suas respectivas versões de software;
3. O gerenciador de atualizações deve fornecer uma interface de linha de comando (CLI) que permita ao usuário interagir com o sistema;

4. O gerenciador de atualizações, no processo de atualização do software em um dispositivo IoT, deve conectar na porta 1234/TCP de cada dispositivo IoT ativo e com versão de software inferior à versão mais recente, e enviar a nova versão do software. O dispositivo IoT deve então substituir a versão antiga pela nova;

5. Ao tentar conectar a um dispositivo IoT para enviar a atualização, o gerenciador de atualizações pode se deparar com dispositivos que não estão mais ativos. Nesse caso, o gerenciador deve registrar uma mensagem de erro no console, remover o dispositivo da lista de ativos e continuar com o próximo dispositivo da lista;

6. Monte um cenário de teste com pelo menos três dispositivos IoT e um gerenciador de atualizações. Você deve executar cada dispositivo IoT e o gerenciador de atualizações em contêineres Docker separados, mas todos na mesma rede Docker.

7. Ao iniciar, o dispositivo IoT deve aceitar a versão inicial do software como um argumento de linha de comando. Por exemplo, `java -jar dispositivo-iot.jar v1.0` inicia o dispositivo IoT com a versão do software `v1.0`.

8. O gerenciador de atualizações deve sempre iniciar com a versão do software definida como `v1.0` e o usuário pode incrementar essa versão através da interface CLI;

9. No arquivo `Readme.md` você deve apresentar o passo a passo, que deve ser claro o suficiente para que qualquer pessoa consiga reproduzir o cenário de teste que você montou, para subir os contêineres e como executar os dispositivos IoT e o gerenciador;

10. No arquivo `Readme.md` inclua as capturas de tela que demonstrem o funcionamento do sistema, incluindo a saída do gerenciador de atualizações mostrando a lista de dispositivos IoT ativos, as versões do software e as mensagens de erro quando um dispositivo não responde durante a atualização;

11. No arquivo `Readme.md` você deverá indicar quais funcionalidades atendeu e quais não atendeu.

Plágio não é tolerado

Você deve ser o único(a) responsável por fazer a entrega para essa atividade. Todo o código ou texto deverá ser produzido exclusivamente por você, exceto trechos de códigos que possam ter sido fornecidos como parte do enunciado.

Você pode discutir com outros estudantes com o intuito de esclarecer pontos, isso é até incentivado, porém você não poderá copiar trechos de códigos, textos ou soluções de qualquer fonte (e.g. colegas da mesma turma ou de turmas anteriores, repositórios de códigos na Internet ou soluções providas por serviços como Copilot e ChatGPT).

Documento licenciado sob Creative Commons “Atribuição 4.0 Internacional”.