

Problema 1: Internet das Coisas

Calendário

Semana	Data	Grupo Tutorial 1 e 3	Grupo Tutorial 2
1	02/04	Sessão 1: Apresentação do Problema	
	04/04	Sessão 3: Tutorial	Sessão 3: Tutorial
2	09/04	Sessão 4: Desenvolvimento	Sessão 4: Tutorial
	11/04	Sessão 5: Tutorial	Sessão 5: Desenvolvimento
3	16/04	Sessão 6: Desenvolvimento	Sessão 6: Tutorial
	18/04	Feriado de Micareta	
4	23/04	Sessão 5: Tutorial	Sessão 5: Desenvolvimento
	25/04	Sessão 6: Desenvolvimento	Sessão 6: Tutorial
5	02/05	Sessão 7: Tutorial	Sessão 7: Desenvolvimento
	04/05	Sessão 8: Desenvolvimento	Sessão 8: Desenvolvimento
6	09/05	Apresentação do resultado e entrega do produto	
	11/05		

Contexto

A Internet das Coisas (*Internet of Things*, IoT) é um termo criado por Kevin Ashton, pesquisador britânico que concebeu um sistema de sensores onnipresentes e um padrão global para RFID, permitindo assim identificar objetos e conectar informações sobre o mundo físico à Internet. A proposta de agregar os sistemas *cyber* físicos (*Cyber-Physical System*, CPS) à Internet surgiu antes de Ashton sugerir o termo IoT em 1999, mas foi apenas na última década que esta tecnologia ganhou popularidade devido aos avanços em diferentes áreas como sistemas embarcados, microeletrônica, comunicação e sensoriamento. Recentemente, a IoT tem recebido bastante atenção tanto da academia quanto da indústria devido ao seu potencial em áreas como saúde, energia, cidades inteligentes, etc. Embora as "coisas" e a conectividade entre elas através da Internet sejam os principais ingredientes de uma arquitetura IoT, o seu valor principal está no preenchimento de lacunas e demandas entre o mundo físico e o mundo digital através de sistemas ou aplicações que utilizem os dados gerados por sensores para automatização de sistemas e processos benéficos a sociedade.

Problema

Neste problema, você faz parte de uma *startup* na área de IoT que acabou de ser adquirida por uma empresa que desenvolve dispositivos para variados setores da indústria. Embora essa empresa venha adicionando capacidades de rede aos dispositivos que ela produz, ela ainda não desenvolveu a comunicação entre esses dispositivos e suas diferentes aplicações. A motivação para aquisição da *startup* por essa empresa foi exatamente o *know-how* de sua equipe no desenvolvimento de *middleware* distribuído. Dessa forma, a sua equipe foi direcionada para o desenvolvimento de um serviço que permita facilitar a comunicação entre os diferentes dispositivos e as aplicações que precisam dos dados gerados por esses dispositivos e ou comandar seu comportamento. Ficou decidido que o sistema deve ser implementado através de um serviço *broker* que permita a troca de mensagens entre os clientes do serviço (dispositivos e aplicações) utilizando como base o subsistema de rede TCP/IP. Através desse serviço, um dispositivo ou aplicativo deve ser capaz de enviar e ou receber mensagens para outros dispositivos ou aplicações interessadas. Assim, os diferentes dispositivos e aplicações podem se comunicar e trabalhar de forma cooperativa.

Produto

Além do serviço *broker* que permita a troca de mensagens entre os clientes do serviço, os seguintes componentes do produto devem ser implementados e apresentados pelo aluno:

1. Um dispositivo virtual (sensor e atuador) simulado através de software (programa em qualquer linguagem) para geração de dados fictícios, onde:
 - a. O dispositivo deve usar uma abordagem não confiável para o envio de dados e uma abordagem confiável para os comandos de gerenciamento;
 - b. O dispositivo virtual deve possuir uma interface por parâmetros (não gráfica) para definir a geração dos dados em tempo real. Por exemplo, para um sensor de temperatura, deve existir um parâmetro ou comando para definir e ou alterar a sua temperatura. O mesmo pode ser aplicado para parâmetros de controle como ligar, pausar, desligar, etc.;
2. Uma aplicação simples com interface (gráfica ou não) que permita a seleção remota de um dispositivo e o seu controle (ligar, reiniciar, desligar, etc.), bem como a visualização dos dados gerados;
3. Uma API RESTful oferecida pelo serviço *broker* que permita a troca de dados e comandos entre a aplicação e os dispositivos emulados.

Restrições

- Os componentes *backend* do sistema (*dispositivos* e *broker*) devem ser desenvolvidos e testados por meio de contêineres Docker para facilitar a execução de mais de uma instância;
- O dispositivo virtual deve implementado usando a interface de socket nativa do TCP/IP, sem uso da arquitetura REST, pois os sensores da empresa não possuem capacidade para rodar servidores HTTP;
- Por questões comerciais, nenhum framework de troca de mensagens deve ser usado para implementar a comunicação entre os dispositivos virtuais e o serviço *broker*. Porém, frameworks para construção de API REST podem ser usados para implementar a comunicação entre o serviço *broker* e a aplicação;
- A API REST suportada pelo serviço *broker* deve ser projetada e testada usando softwares como Insomnia ou Postman para garantir sua compatibilidade com a arquitetura REST.

Nossas Regras

- Os alunos devem implementar o problema individualmente;
- O prazo final de entrega do trabalho e apresentação será dia 09/05/2023;
- O código fonte deve ser entregue devidamente comentado (relatório) por meio da plataforma GitHub;
- Neste problema, será estabelecida uma agenda de apresentação sequencial onde o aluno terá 20 minutos para executar e realizar a sua apresentação;
- O sistema deve ser desenvolvido, testado e apresentado no Laboratório de Redes e Sistemas Distribuídos (LARSID), onde, durante a apresentação, o tutor realizará arguições ao aluno.
- O aluno deve apresentar a interface do sistema funcionando e ser arguido sobre os aspectos de implementação relacionados.

Observações

- Trabalhos entregues fora do prazo serão penalizados com 20% do valor da nota + 5% por dia de atraso, dentro da mesma semana da entrega final;
- Trabalhos copiados da INTERNET ou de qualquer outra fonte e trabalhos idênticos terão nota ZERO;
- As informações para solução do problema podem ser ALTERADAS no decorrer das sessões.

Avaliação

A nota final será a composição de 03 notas:

1. Desempenho individual (30%);
2. Relatório do Sistema (no GitHub) (20%);
3. Produto (código incluso) (50%).