

Especificação do projeto: cliente/servidor em internet das coisas

Prof. Felipe Duque
`felipe.duque@ufpe.br`

- O projeto deve ser escrito em linguagem C ou C++;
- O projeto vale 4 pontos na nota final;
- O projeto é em dupla;
- A avaliação do projeto será baseada na sua funcionalidade (nota referente à dupla) e numa arguição oral entre o professor e cada integrante de cada dupla (nota individual);
- A avaliação será privativa entre a dupla e o professor;
- Aproveite as aulas dedicadas ao acompanhamento do projeto;
- Dúvidas? `felipe.duque@ufpe.br`
- **Cada grupo deverá implementar ambos: servidor e cliente.**

Numa aplicação de internet das coisas (IoT, *internet of things*), uma rede de sensores se comunica com uma central C (que tem o papel de um servidor). Cada sensor S (que tem o papel de um cliente) é alimentado por uma bateria, cuja carga se esvai com o tempo. Periodicamente, a cada **ciclo** segundos, a central envia uma requisição a S , com o objetivo de saber se o sensor está “vivo”. Caso o sensor responda afirmativamente, a central lhe envia outra requisição, agora com o objetivo de saber o nível atual da bateria do sensor. Assim que S recebe essa requisição, ele realiza uma medição do nível atual da bateria. Caso o nível recebido por C seja menor do que o

limiar `limiar`, a central mostra ao usuário um alerta. Como o sistema pode ser composto por vários sensores, *C* precisa garantir que os valores recebidos sejam processados na ordem de chegada. Logo, cada valor recebido de cada sensor é armazenado numa fila em *C*. Assim que um valor é processado (i.e., é comparado com `limiar`), ele é removido da fila.

O projeto consiste na implementação desse sistema com uma central e um sensor.

O funcionamento do sensor *S* é destrinchado a seguir:

1. *S* se conecta a *C*;
2. *S* permanece “escutando” o *socket* continuamente, à espera de requisições de *C*;
3. Quando *S* recebe de *C* uma requisição de “vivo?”, *S* envia a *C* uma mensagem afirmativa;
4. Quando *S* recebe de *C* uma requisição de “carga?”, *S* envia a *C* um valor do tipo `float` entre 0 e 1, indicando o nível atual da carga;
5. O nível da carga é reduzido continuamente.

O funcionamento da central *C* é destrinchado a seguir:

1. *C* inicia a operação e lê do usuário os valores `ciclo`, em segundos, e `limiar`, do tipo `float`, entre 0 e 1;
 - (a) **Vale pontuação extra:** faça os valores serem lidos a partir de um arquivo, de forma amigável ao usuário. Será necessário realizar um *parse* do arquivo.
2. *C* espera, indefinidamente, algum cliente se conectar;
3. Assim que um cliente *S* se conectar, *C* envia a *S*, a cada `ciclo` segundos, uma requisição de “vivo?” e espera a resposta de *S*;
 - (a) A resposta nunca será negativa;

-
- (b) Não é obrigatório considerar o caso em que S não responde;
 - (c) **Vale pontuação extra:** considere o caso em que, se S não responder em até **timeout** segundos, a central emitirá um alerta ao usuário.
4. Ao receber uma resposta positiva de S , C envia a S uma requisição de “carga?” e espera a resposta de S ;
 5. C recebe o valor da carga, que é do tipo **float**, e o armazena numa estrutura de fila;
 6. Quando a fila atinge o tamanho de 5 elementos, C processa cada valor que consta na fila: se for menor do que **limiar**, é emitida uma mensagem de alerta ao usuário;
 7. Cada valor processado é removido da fila.