



UNIVERSIDADE
FEDERAL DO CEARÁ

UNIVERSIDADE FEDERAL DO CEARÁ

Outubro de 2019

Curso: Bacharelado em Ciência da Computação

Semestre: 2019.2

Disciplina: Sistemas Distribuídos

Professor: Filipe Fernandes dos Santos Brasil de Matos

Nome da dupla: _____

TRABALHO PRÁTICO SOCKET/PROTOBUF

O objetivo deste trabalho é implementar uma aplicação de Agenda simples utilizando algumas tecnologias de comunicação remota inter-processo. A aplicação **DEVE** usar uma arquitetura Cliente/Servidor, onde a parte Servidora deverá gerenciar a Agenda do usuário (Cliente). O Cliente deve interagir com o Servidor utilizando algum tipo de interface (Gráfica ou não). O Servidor **DEVE**, pelo menos, disponibilizar um CRUD dos Contatos da Agenda. Algumas outras características da aplicação:

1. A Agenda deve ser tratada como uma lista de Contatos;
2. A Agenda deve ser persistida em disco utilizando alguma ferramenta a ser escolhida pelos membros. Pode ser um banco de dados relacional, não-relacional, arquivo de texto...
3. Cada Contato **DEVE** ser modelado com os campos: Nome, Endereço Completo, Telefone(s) e E-mail. O campo Telefone(s) é multivalorado, ou seja, pode assumir de um a N valores. Desses campos, somente o Nome e Telefone(s) são obrigatórios (os demais são opcionais).

As tecnologias/protocolos de comunicação utilizados serão os seguintes:

1. **Socket:** A equipe deve realizar a comunicação usando sockets puro disponibilizados pelo Sistema Operacional Linux e também deverão definir um protocolo de comunicação simples entre eles, ou seja, formato das mensagens a serem trocadas entre os sockets, bem como o comportamento das partes Cliente e Servidora antes, durante e depois do envio das mensagens. Para este cenário, **DEVE-SE OBRIGATORIAMENTE** utilizar a linguagem de programação C. Essa parte do trabalho valerá de 0 a 3 pontos. **Serão acrescidos 0.5 ponto extra caso a equipe desenvolva tanto o Cliente, como o Servidor utilizando *threads*: 2 para o Cliente e 2 para o Servidor¹.**
2. **Protobuf:** A equipe deve implementar as mesmas funcionalidades do aplicação desenvolvida no cenário Socket, porém, agora utilizando o Protobuf como formato padrão de mensagem. Para este cenário, **PODE-SE UTILIZAR QUALQUER LINGUAGEM DE PROGRAMAÇÃO²**. Para mais informações sobre o Protobuf consultar <https://developers.google.com/protocol-buffers>. Essa parte do trabalho valerá de 0 a 2 pontos. **Serão acrescidos 0.5 ponto extra caso a equipe desenvolva Cliente e Servidor utilizando linguagens de programação distintas.**

A entrega do trabalho será feita por todos os grupos até as 23:59:59 do dia 01 de Novembro de 2019 (Sexta-Feira). Na entrega, um membro da equipe deverá enviar os código-fontes e arquivos necessários para a execução da aplicação (em .zip's distintos: socket.zip e protobuf.zip) para o meu

e-mail institucional (filipe.fernandes@crateus.ufc.br). O e-mail deverá ter como título **Trabalho Prático de Socket/Protobuf** e o corpo do e-mail deverá conter: **I) os nomes dos integrantes da equipe, II) um minitutorial de como executar os programas criados.**

Além disso, cada grupo deverá apresentar os seus dois projetos para mim no dia 04 de Novembro de 2019 (Segunda-Feira). Nesse dia, todos os participantes do grupo deverão estar presentes (participantes ausentes ficarão sem a nota do trabalho) e deverão, junto comigo, colocar o sistema para rodar, fazer alguns testes. Os integrantes também deverão apresentar os códigos gerados e, eventualmente, responder alguns questionamentos meus sobre a implementação. A ordem das apresentações será definida por sorteio e os horários serão definidos diretamente com cada equipe.

Todos os trabalhos serão analisados e comparados. Caso seja identificada cópia de trabalhos, todos os trabalhos envolvidos receberão nota ZERO. Trabalhos com atraso serão aceitos mediante um desconto de 1.0 ponto na nota final por dia de aula de atraso. Este trabalho deverá ser feito em grupo de, no máximo, três componentes.

Qualquer dúvida me procurem no Gabinete 2 ou me contactem via e-mail.

¹ No lado Cliente, uma *thread* para interagir com o usuário, outra para interagir com o Servidor. Ambas se comunicando via buffer. No lado do Servidor, uma thread para receber as requisições do Cliente e outra para tratar a requisição. Estas threads também devem se comunicar utilizando um buffer intermediários.

² Atenção as linguagens suportadas pelo Protobuf.