Universidade Federal do Ceará (UFC) - Campus Quixadá
Disciplina: Sistemas Distribuídos - Período 2015.2
Profª. Atrícia Sabino - atriciasabino@gmail.com
Avaliação Parcial Prática 2

(1) Crie uma aplicação, Cliente/Servidor usando o serviço **RPC** (chamada de procedimento remoto). **Não crie sockets nessa questão!**

- Uma aplicação com no mínimo 4 métodos.
- Chame todos os métodos no cliente - "cliente magro", o servidor implementará esses métodos.
- Deve ser usado obrigatoriamente o xml-rpc.
- Cliente e Servidor deverá ser executado em máquinas diferentes.

(2) Use o protocolo Buffer para criar um **serviço/aplicação**:

- No mínimo três métodos na aplicação.
- Defina as mensagens no formato **.proto.**
- Use o compilador **protoc** para geração automática de código.
- Use a API **Python** Protocol Buffer para escrever e ler mensagens.
    a. Empacotar (marshalling) e Desempacotar (unmarshalling) mensagens

(3) Use o método **.gethostbyname** para conectar-se a algum servidor ftp online.

- A conexão deverá ser com usuário anônimo.
- A saída do programa deverá ser o IP do FTP e a mensagem de conectado ao servidor.
- Exiba a porta do serviço FTP.

(4) Use o s**ocket.gethostbyname** para resolução de nomes.

- Leia uma lista de Domínios em um arquivo,
- Resolva o nome e retorne o ip.
- Se o nome do host não existe, defina um tempo em segundos para exibir uma mensagem de **DNS timeout**.

(5) Sobre a programação em socket, crie um socket **TCP** servidor de arquivos**.**

- Rode o cliente passando como argumento o servidor, porta e seu nome. **Ex: python socket.py localhost 5000 fulano**

- Seu servidor deve ser multithread, execute vários clientes para verificar se o servidor atende a todos.

Universidade Federal do Ceará (UFC) - Campus Quixadá
Disciplina: Sistemas Distribuídos - Período 2015.2
Profª. Atrícia Sabino - atriciasabino@gmail.com
Avaliação Parcial Prática 2

- Em seguide adicione uma função para que o cliente solicite arquivos para baixar do servidor. O servidor deve retornar o arquivo ou uma mensagem de arquivo não encontrado.

(6) Sobre a reprensentação externa de dados, e serealização. Crie um socket Cliente/Servidor com uma aplicação com no mínimo dois métodos.

- O cliente deve empacotar a mensagem de **requisição** antes de enviar para o servidor. E desempacotar a mensagem de **reply** enviada pelo servidor**.**

- O servidor deve desempacotar a mensagem de **requisição** do cliente, e empacotar o reply e enviar para o cliente.

(7) Use o código de multicast da aula prática para criar um chat multicast.

- Usar um timeout para sair da **conversa,** caso não digite nada.
- Sair do **grupo** multicast, caso a conversa não esteja interessante.

(8) Baseado na questão anterior do chat **multicast,** adicione ao código uma função de:

- Login.
- Logoff.
- Mensagens enviadas para os usuários do grupo via mensões, por exemplo **@alguém.**
- Função para mostrar usuários online.