Grupos de até 4 alunos

O objetivo deste trabalho é implementar uma versão simplificada de um aplicativo de comunicação por mensagens similiar ao Whatsapp. A implementação deve seguir um modelo *cliente-servidor* e utilizar o *protocolo TCP* para troca das mensagens. Além disso, as seguintes ações devem ser suportadas.

Registrar cliente: processo realizado na primeira vez que um cliente conecta ao serviço. O cliente enviará uma mensagem ao servidor indicando a intenção de criar uma identificação no sistema. Como resposta o servidor enviará uma mensagem com o código único gerado para o usuário. Após esta etapa o cliente pode já ficar conectado ao servidor ou não.

Mensagem enviada pelo cliente: 01
Resposta do servidor: 02**********

Os dois primeiros caracteres da mensagem representam o seu código (01 para registrar, 02 resposta do pedido de registro). A resposta do servidor contém uma sequência de 13 dígitos que serão o identificador único daquele cliente.

Conectar cliente: processo realizado pelo cliente para se conectar ao servidor, seja pela primeira vez ou após alguma desconexão. Quando um cliente se conecta, o servidor deve verificar se existem mensagens pendentes para o cliente e realizar a sua entrega (mais detalhes em breve).

Mensagem enviada pelo cliente: 03********** onde 03 é o código de mensagem seguido por 13 digitos que representam o identificador único do cliente

Enviar mensagem: processo realizado por um cliente (originador) para enviar mensagem para outro cliente (destinatário). Segue o seguinte formato *cod(2)src(13)dst(13)timestamp(10)data(218)* onde:

- cod são dois dígitos indicando o tipo de mensagem, neste caso, 05
- src é uma sequência de 13 dígitos representando o originador da mensagem
- dst é uma sequência de 13 dígitos representando o destinatário da mensagem
- timestamp é a data e hora de envio da mensagem em formato POSIX
- data são até 218 caractéres de conteúdo que podem ser enviados

Ao receber uma mensagem deste tipo, o servidor terá um dos seguintes comportamentos:

1. Se o destinatário estiver online, realizar a entrega da mensagem de forma imediata
2. Se o destinatário não estiver online, armazenar a mensagem internamente e realizar sua entrega no momento que o cliente reconectar

A entrega da mensagem por parte do servidor a um cliente segue o mesmo formato descrito acima, sendo a única diferença o seu código, que será 06

Confirmação de entrega de mensagem para o servidor. Por padrão, o protocolo TCP gerará uma confirmação de entrega da mensagem. Esta pode ser utilizada para sinalizar ao cliente originador da mensagem que ela foi entregue com sucesso ao servidor da aplicação.

Confirmação de entrega de mensagem para o cliente. Quando o servidor conseguir entregar a mensagem para o cliente (destinatário), ele deverá notificar o originador sobre a entrega. Isto será feito pelo envio de uma mensagem com o seguinte formato: cod(2)dest(2),timestamp(10) onde:

- cod são dois dígitos indicando o tipo de mensagem, neste caso, 07
- dst é uma sequência de 13 dígitos representando o identificador do cliente que recebeu a mensagem
- timestamp é a data e hora, em formato POSIX, da última mensagem entregue pelo servidor originada por um cliente e entregue para um destinatário

Ao receber uma mensagem deste tipo, o cliente deverá assumir que todas as mensagens enviadas para o cliente (dst) até o momento (timestamp) foram corretamente entregues pelo servidor

Confirmação de leitura da mensagem pelo cliente. Quando o cliente efetivamente realizar a leitura de mensagens, o servidor deverá ser notificado para que possa notificar o originador. Assim, ao abrir uma mensagem, um cliente deverá enviar ao servidor uma mensagem com o seguinte formato: cod(2)src(13)timestamp(10) onde:

- cod são dois dígitos indicando o tipo de mensagem, neste caso, 08
- src é uma sequência de 13 dígitos representando o identificador do cliente que originou a mensagem que foi lida
- timestamp é a data e hora, em formato POSIX, da última mensagem lida pelo cliente

Ao receber uma mensagem deste tipo, o servidor deverá gerar uma mensagem informando o cliente originador sobre a leitura da mensagem. A mensagem tem o formato cod(2)dst(13)timestamp(10) onde:

- cod são dois dígitos indicando o tipo de mensagem, neste caso, 09
- dst é uma sequência de 13 dígitos representando o identificador do cliente que leu a mensagem
- timestamp é a data e hora, em formato POSIX, da última mensagem lida pelo destinatário

Criar grupo. Consiste no processo em criar um grupo contendo múltiplos identificadores de cliente. O processo consiste um cliente enviar uma mensagem para o servidor contendo as informações necessárias no seguinte formato cod(2)criador(13)timestamp(10)members(7*13), onde:

- cod são dois dígitos indicando o tipo de mensagem, neste caso, 10
- criador é uma sequência de 13 dígitos representando o identificador do cliente que criou o grupo
- timestamp é a data e hora, em formato POSIX, da solicitação de criação do grupo
- members é uma sequência de até 7 identificadores de 13 digitos indicando os demais membros do grupo

Ao receber esta mensagem o servidor deverá gerar um identificador único do grupo bem como notificar os membros (e o criador) que eles fazem parte agora de um grupo. Isto é feito através de uma mensagem no seguinte formato cod(2)id(13)timestamp(10)membros(8*13) onde:

- cod são dois dígitos indicando o tipo de mensagem, neste caso, 11
- id é um identificador único de 13 dígitos associado ao grupo
- timestamp é a data e hora, em formato POSIX, da criação do grupo
- members é uma sequência de até 8 identificadores de 13 digitos indicando os demais membros do grupo

Observações.

- O cliente deverá de alguma forma armazenar o seu próprio ID, a sua lista de contatos e grupos dos quais faz parte, além do histórico de mensagens trocadas. Estas informações devem ser persistentes, isto é, se o cliente for encerrado, quando ele for reexecutado, deverá restaurar o estado anterior ao encerramento.
- Para facilitar, o cliente pode ter um método para cadastrar contatos em sua lista de contatos.
- O servidor deve armazenar as mensagens ainda não entregues. Estas devem ser entregues no momento que o cliente se conectar. A forma que as mensagens ficarão armazenadas pode ser definida por cada grupo.
- Todas as mensagens devem ser codificadas em UTF-8 para garantir a consistência das implementações.
- A implementação devera ser feita em Python
- As mensagens são sequencias de caracteres