

WHATSPROG – INTERCAMBIADOR DE MENSAGENS
PROFESSOR: ADELARDO ADELINO DANTAS DE MEDEIROS

DESCRIÇÃO

O objetivo é desenvolver em C++ um aplicativo cliente-servidor, denominado WhatsProg, capaz de trocar mensagens de texto entre usuários em máquinas diferentes. O servidor se conecta com todos os clientes dos usuários e encaminha as mensagens para os destinatários corretos. A aplicação utilizará threads e sockets TCP. A comunicação entre cliente e servidor deverá utilizar a porta 23456.

Todo usuário, antes de utilizar o WhatsProg, deverá se cadastrar no servidor com um login e senha. O login deve ser único e ter entre 6 e 12 caracteres. A senha também deve ter entre 6 e 12 caracteres. Tanto no login quanto na senha os caracteres maiúsculos e minúsculos são diferentes.

Cada mensagem enviada pode ter no máximo 255 caracteres. O destinatário deve ser um usuário previamente cadastrado: caso não exista, a mensagem é rejeitada. As mensagens destinadas a um usuário que esteja conectado devem ser transmitidas imediatamente. Caso o usuário esteja desconectado, a mensagem deve ser armazenada em um buffer de mensagens. Quando o usuário se conectar, todas as mensagens armazenadas devem ser transmitidas e eliminadas do buffer.

Quando uma mensagem é lida (ou seja, visualizada em tela pelo usuário), uma notificação é enviada do cliente do destinatário para o cliente do remetente, via servidor. Também são enviadas, pelo servidor, notificações quando a mensagem é recebida e quando é transmitida para o destinatário. O programa cliente exibe para o usuário a situação das mensagens que foram enviadas por ele:

- MSG_ENVIADA (?): a mensagem foi enviada para o servidor.
- MSG_RECEBIDA (✓): o servidor confirmou a recepção.
- MSG_ENTREGUE (✓✓): a mensagem foi entregue (transmitida) ao destinatário pelo servidor.
- MSG_LIDA (✓✓✓): a mensagem foi lida (visualizada) pelo usuário no cliente destinatário.

As comunicações entre o cliente e o servidor, ou vice-versa, seguem um padrão:

1. Os primeiros 4 bytes sempre contêm um inteiro (`int32_t`) que indica o comando (ver lista de comandos a seguir)
2. Em seguida, seguem os parâmetros do comando, caso existam (cada comando tem um número diferente de parâmetros).

Os parâmetros que não têm tamanho fixo (uma string, por exemplo) seguem o mesmo padrão:

- 2.1. Inicialmente, é transmitido um inteiro de 4 bytes (`int32_t`) contendo o número N de bytes do parâmetro
- 2.2. Em seguida, são transmitidos N bytes com o parâmetro propriamente dito.

[OPCIONAL] O programa servidor, ao encerrar, deve salvar as informações em arquivo, de tal forma que, ao ser lançado novamente, esteja no mesmo estado anterior. Devem ser armazenados:

- login e senha de todos os usuários cadastrados.
- todas as mensagens cujo processamento ainda não foi concluído (ou seja, estão no buffer).

COMANDOS

NOME	VALOR	PAR 1	PAR 2	PAR 3	SIGNIFICADO (*)
CMD_NEW_USER	1001	login: string	senha: string	-	Criação de um novo usuário (C→S)
CMD_LOGIN_USER	1002	login: string	senha: string	-	Conexão com um usuário já existente (C→S)

NOME	VALOR	PAR 1	PAR 2	PAR 3	SIGNIFICADO (*)
CMD_LOGIN_OK	1003	-	-	-	Conexão OK (S→C)
CMD_LOGIN_INVALIDO	1004	-	-	-	Conexão inválida (S→C)
CMD_NOVA_MSG	1005	Id: int32_t	usuário: string	texto: string	Nova mensagem (Cr→S e S→Cd)
CMD_MSG_RECEBIDA	1006	Id: int32_t	-	-	Mensagem recebida pelo servidor (S→Cr)
CMD_MSG_ENTREGUE	1007	Id: int32_t	-	-	Mensagem entregue ao destinatário (S→Cr)
CMD_MSG_LIDA1	1008	Id: int32_t	usuário: string	-	Mensagem visualizada pelo destinatário (Cd→S)
CMD_MSG_LIDA2	1009	Id: int32_t	-	-	Mensagem visualizada pelo destinatário (S→Cr)
CMD_ID_INVALIDA	1010	Id: int32_t	-	-	ID da mensagem inválida (S→Cr)
CMD_USER_INVALIDO	1011	Id: int32_t	-	-	Usuário (remetente) da mensagem inválido (S→Cr)
CMD_MSG_INVALIDA	1012	Id: int32_t	-	-	Texto da mensagem inválido (S→Cr)
CMD_LOGOUT_USER	1013	-	-	-	Desconexão do usuário (C→S)

(*) Sentido de envio: S=Servidor, C=Cliente, Cr=Cliente remetente, Cd=Cliente destinatário

ESTADOS (STATUS) DAS MENSAGENS NOS BUFFERS

A – No cliente

ESTADO ANTERIOR	EVENTO	AÇÃO A SER REALIZADA	NOVO ESTADO
-	Mensagem digitada	Criar conversa, se necessário Inserir no buffer da conversa Enviar CMD_NOVA_MSG	MSG_ENVIADA
-	Recebe CMD_NOVA_MSG	Criar conversa, se necessário Inserir no buffer da conversa Exibir aviso de chegada (alteração no número de mensagens da conversa) Exibir mensagem, se conversa estiver visualizada	MSG_ENTREGUE
MSG_ENVIADA (remetente)	Recebe CMD_MSG_RECEBIDA	Exibir status (✓), se conversa estiver visualizada	MSG_RECEBIDA
	Recebe CMD_MSG_ENTREGUE	Exibir status (✓), se conversa estiver visualizada	MSG_ENTREGUE
	Recebe CMD_MSG_LIDA2	Exibir status (✓), se conversa estiver visualizada	MSG_LIDA
	Recebe CMD_ID_INVALIDA Recebe CMD_USER_INVALIDO Recebe CMD_MSG_INVALIDA	Remover do buffer Exibir mensagem de erro	-
MSG_RECEBIDA (remetente)	Recebe CMD_MSG_ENTREGUE	Exibir status (✓), se conversa estiver visualizada	MSG_ENTREGUE

ESTADO ANTERIOR	EVENTO	AÇÃO A SER REALIZADA	NOVO ESTADO
	Recebe CMD_MSG_LIDA2	Exibir status (✓), se conversa estiver visualizada	MSG_LIDA
MSG_ENTREGUE (remetente)	Recebe CMD_MSG_LIDA2	Exibir status (✓), se conversa estiver visualizada	MSG_LIDA
MSG_ENTREGUE (destinatário)	Conversa visualizada	Enviar CMD_MSG_LIDA1	MSG_LIDA

B – No servidor

ESTADO ANTERIOR	EVENTO	AÇÃO A SER REALIZADA	NOVO ESTADO
-	Recebe CMD_NOVA_MSG do remetente	Inserir no buffer de mensagens pendentes Enviar CMD_MSG_RECEBIDA p/ remetente Se possível, enviar CMD_NOVA_MSG p/ destinatário	MSG_RECEBIDA
MSG_RECEBIDA	Envia CMD_NOVA_MSG p/ destinatário	Se possível, enviar CMD_MSG_ENTREGUE p/ remetente	MSG_ENTREGUE
MSG_ENTREGUE	Recebe CMD_MSG_LIDA1	Se possível, enviar CMD_MSG_LIDA2 p/ remetente	MSG_LIDA
MSG_LIDA	Envia CMD_MSG_LIDA2 p/ remetente	Remover do buffer	-

CASOS DE USO

A – Cadastro de novo usuário

- O usuário digita login e senha.
- O cliente se conecta ao servidor e envia CMD_NEW_USER (Caso de uso J), com parâmetros:
 - login: string
 - senha: string
- O servidor testa se o novo usuário é válido e se já não existe outro com mesmo login:
 - Se for válido:
 - Cadastra o novo usuário.
 - Conecta o cliente, associando com o socket criado para a conexão.
 - Responde CMD_LOGIN_OK (Caso de uso I), sem parâmetros.
 - O cliente pode começar a utilizar a conexão para troca de mensagens.
 - Se não for válido:
 - O servidor responde CMD_LOGIN_INVALIDO (Caso de uso I), sem parâmetros.
 - O servidor fecha a conexão.
 - O cliente deverá fazer nova conexão.

B – Conexão de usuário já existente

- O usuário digita login e senha.
- O cliente se conecta ao servidor e envia CMD_LOGIN_USER (caso de uso J), com parâmetros:
 - login: string
 - senha: string

3) O servidor testa se o usuário é válido:

- usuário está cadastrado;
 - a senha está correta; e
 - este usuário já não está conectado em outro socket.
- a) Se for válido:
- i) O servidor conecta o cliente, associando com o socket criado para a conexão.
 - ii) O servidor responde `CMD_LOGIN_OK` (Caso de uso I), sem parâmetros.
 - iii) Se for o caso, o servidor envia para o cliente todas as eventuais mensagens que sejam destinadas a ele e que estejam armazenadas com status `MSG_RECEBIDA` (Casos de uso D e I), alterando o status dessas mensagens para `MSG_ENTREGUE`.
 - iv) Se for o caso, o servidor envia para o cliente todas as eventuais confirmações de leitura de mensagens que foram enviadas por ele e que estejam armazenadas com status `MSG_LIDA` (Casos de uso G e I), removendo em seguida essas mensagens do buffer.
 - v) O cliente pode começar a utilizar a conexão para troca de mensagens.
- b) Se não for válido:
- i) O servidor responde `CMD_LOGIN_INVALIDO` (Caso de uso I), sem parâmetros.
 - ii) O servidor fecha a conexão.
 - iii) O cliente deverá fazer nova conexão.

C – Envio de mensagem (do cliente remetente para o servidor)

- 1) O usuário escolhe uma conversa (ou cria uma nova conversa, caso ainda não exista uma conversa associada ao destinatário) e digita o texto de uma mensagem.
- 2) O cliente remetente armazena a mensagem no buffer da conversa associada ao destinatário, com dados:
 - Id
 - remetente (o próprio usuário) e destinatário (o usuário da conversa)
 - texto
 - status `MSG_ENVIADA`.
- 3) O cliente remetente envia `CMD_NOVA_MSG` para o servidor (Caso de uso J), com parâmetros:
 - Id única da mensagem: `int32_t`
 - Nome do usuário (destinatário): `string`
 - Texto da mensagem: `string`
- 4) O servidor testa se a mensagem está correta:
 - Id é maior que a última recebida daquele cliente.
 - Destinatário é um usuário cadastrado.
 - Texto obedece às restrições de tamanho.

a) Se estiver correta:

 - i) O servidor armazena a mensagem no buffer de mensagens pendentes, com dados:
 - Id
 - remetente e destinatário
 - texto
 - status `MSG_RECEBIDA`.
 - ii) O servidor envia `CMD_MSG_RECEBIDA` para o cliente remetente, com parâmetro Id (Caso de uso I).
 - iii) O cliente remetente altera o status da mensagem para `MSG_RECEBIDA`.
 - iv) Caso o destinatário esteja conectado, o servidor envia a mensagem para o cliente destinatário (Casos de uso D e I).

b) Se não estiver correta:

 - i) O servidor responde para o cliente remetente (Caso de uso I):

- Se o erro foi na Id, `CMD_ID_INVALIDA`, com parâmetro Id.
 - Se o erro foi no destinatário, `CMD_USER_INVALIDO`, com parâmetro Id.
 - Se o erro foi no texto, `CMD_MSG_INVALIDA`, com parâmetro Id.
- ii) O cliente remetente emite uma mensagem de erro e remove a mensagem.

D – Envio de mensagem (do servidor para um cliente destinatário)

Esse caso de uso acontece logo após a recepção pelo servidor de uma mensagem destinada a um usuário conectado (continuação do caso C.4.a.iv) ou quando um usuário se conecta e existem mensagens que foram enviadas para ele por outros usuários enquanto ele esteve desconectado (continuação do caso B.3.a.iii).

- 1) O servidor seleciona uma mensagem que:
 - esteja no buffer de mensagens pendentes;
 - tenha o usuário em questão como destinatário; e
 - tenha status `MSG_RECEBIDA`.
- 2) O servidor envia `CMD_NOVA_MSG` para o cliente destinatário (Casos de uso E e I), com parâmetros:
 - Id da mensagem que foi atribuída pelo remetente: `int32_t`
 - Nome do usuário (remetente): `string`
 - Texto da mensagem: `string`
- 3) O servidor altera a mensagem no buffer de mensagens pendentes para o status `MSG_ENTREGUE`.
- 4) Caso o remetente esteja conectado:
 - a) O servidor envia `CMD_MSG_ENTREGUE` para o cliente remetente, com parâmetro Id (Caso de uso I).
 - b) O cliente remetente altera o status da mensagem para `MSG_ENTREGUE`.

E – Recepção de mensagem (pelo cliente destinatário)

- 1) Caso ainda não exista uma conversa associada ao remetente, cria a nova conversa.
- 2) O cliente destinatário armazena a mensagem no buffer da conversa associada ao remetente, com dados:
 - Id
 - remetente (o usuário da conversa) e destinatário (o próprio usuário)
 - texto
 - status `MSG_ENTREGUE`.
- 3) O número de mensagens da conversa é incrementado e exibido (aviso ao usuário que chegou nova mensagem).
- 4) Caso a conversa esteja sendo visualizada no momento, entra na fase de envio de confirmação de leitura para o remetente (Caso de uso F).

F – Visualização de conversa (pelo destinatário)

- 1) Quando uma conversa for visualizada (exibida em tela), o cliente procura todas as mensagens daquela conversa com as seguintes características:
 - O remetente é o usuário da conversa, e não o próprio usuário.
 - O status é status `MSG_ENTREGUE`.
- 2) Para cada mensagem encontrada:
 - a) O cliente destinatário altera a mensagem no buffer correspondente para o status `MSG_LIDA`.

- b) O cliente destinatário envia para o servidor `CMD_MSG_LIDA1` (Caso de uso J), com parâmetros:
 - Id da mensagem: `int32_t`
 - Nome do usuário (remetente): `string`
- c) O servidor altera a mensagem no buffer para o status `MSG_LIDA`.
- d) Caso o remetente esteja conectado, o servidor envia a confirmação de visualização para o cliente remetente (Casos de uso G e I).

G – Envio de confirmação de visualização (do servidor para um cliente remetente)

Esse caso de uso acontece logo após a recepção pelo servidor de uma confirmação de visualização de uma mensagem cujo remetente está conectado (continuação do caso F.2.d) ou quando um usuário se conecta e existem mensagens que foram enviadas por ele e que foram visualizadas pelo destinatário enquanto ele (remetente) esteve desconectado (continuação do caso B.3.a.iv).

- 1) O servidor seleciona uma mensagem que:
 - esteja no buffer de mensagens pendentes;
 - tenha o usuário em questão como remetente; e
 - tenha status `MSG_LIDA`.
- 2) O servidor envia `CMD_MSG_LIDA2` para o cliente remetente, com parâmetro `Id` (Caso de uso I)..
- 3) O servidor remove a mensagem do buffer de mensagens pendentes.
- 4) O cliente remetente altera o status da mensagem para `MSG_LIDA`.

H – Desconectar usuário

- 1) O cliente envia `CMD_LOGOUT_USER` para o servidor (Caso de uso J), sem parâmetros.
- 2) O servidor desloga o cliente associado à conexão e fecha a conexão.

I – Recepção de comando pelo cliente

- 1) Se for `CMD_NEW_USER`:
 - a) Ignorar (erro do servidor).
- 2) Se for `CMD_LOGIN_USER`:
 - a) Ignorar (erro do servidor).
- 3) Se for `CMD_LOGIN_OK`:
 - a) Se for após um pedido de criação ou conexão de usuário:
 - i) Conexão estabelecida corretamente.
 - b) Se não for:
 - i) Ignorar (erro do servidor).
- 4) Se for `CMD_LOGIN_INVALIDO`:
 - a) Se for após um pedido de criação ou conexão de usuário:
 - i) Conexão incorreta: informar ao usuário.
 - b) Se não for:
 - i) Ignorar (erro do servidor).
- 5) Se for `CMD_NOVA_MSG`:
 - a) Se não houver uma conversa associada com o usuário recebido como parâmetro:
 - i) Cria nova conversa.
 - b) Insere nova mensagem na conversa, com status `MSG_ENTREGUE`.
 - c) Incrementa e exibe o número de mensagens da conversa.
 - d) Caso a conversa esteja sendo visualizada:
 - i) Altera o status para `MSG_LIDA`.

- ii) Envia `CMD_MSG_LIDA1` para o servidor.
- 6) Se for `CMD_MSG_RECEBIDA`:
 - a) Testa se existe no buffer de alguma das conversas uma mensagem que:
 - tenha Id igual à que foi recebida como parâmetro;
 - esteja armazenada no buffer tendo o usuário como remetente; e
 - tenha status `MSG_ENVIADA`.
 - i) Se existir:
 - (1) Muda o status da mensagem para `MSG_RECEBIDA`.
 - (2) Exibe para o usuário o novo status da mensagem, caso a conversa esteja sendo visualizada.
 - ii) Caso não exista:
 - (1) Ignorar (erro do servidor ou do cliente)
- 7) Se for `CMD_MSG_ENTREGUE`:
 - a) Testa se existe no buffer de alguma das conversas uma mensagem que:
 - tenha Id igual à que foi recebida como parâmetro;
 - esteja armazenada no buffer tendo o usuário como remetente; e
 - tenha status `MSG_ENVIADA` ou `MSG_RECEBIDA`.
 - i) Se existir:
 - (1) Muda o status da mensagem para `MSG_ENTREGUE`.
 - (2) Exibe para o usuário o novo status da mensagem, caso a conversa esteja sendo visualizada.
 - ii) Caso não exista:
 - (1) Ignorar (erro do servidor ou do cliente)
- 8) Se for `CMD_MSG_LIDA1`:
 - a) Ignorar (erro do servidor)
- 9) Se for `CMD_MSG_LIDA2`:
 - a) Testa se existe no buffer de alguma das conversas uma mensagem que:
 - tenha Id igual à que foi recebida como parâmetro;
 - esteja armazenada no buffer tendo o usuário como remetente; e
 - tenha status `MSG_ENVIADA` ou `MSG_RECEBIDA` ou `MSG_ENTREGUE`.
 - i) Se existir:
 - (1) Muda o status da mensagem para `MSG_LIDA`.
 - (2) Exibe para o usuário o novo status da mensagem, caso a conversa esteja sendo visualizada.
 - ii) Caso não exista:
 - (1) Ignorar (erro do servidor ou do cliente)
- 10) Se for `CMD_ID_INVALIDA`, `CMD_USER_INVALIDO` ou `CMD_MSG_INVALIDA`:
 - a) Testa se existe no buffer de alguma das conversas uma mensagem que:
 - tenha Id igual à que foi recebida como parâmetro;
 - esteja armazenada no buffer tendo eu como remetente; e
 - tenha status `MSG_ENVIADA`.
 - i) Se existir:
 - (1) Remove a mensagem do buffer da conversa.
 - (2) Exibe para o usuário a mensagem de erro apropriada.
 - ii) Caso não exista:
 - (1) Ignorar (erro do servidor ou do cliente)
- 11) Se for `CMD_LOGOUT_USER`:
 - a) Ignorar (erro do servidor).
- 12) Se for um comando desconhecido:
 - a) Ignorar (erro do servidor).

J – Recepção de comando pelo servidor

- 1) Se for `CMD_NEW_USER`:
 - a) Testa se o cliente associado à conexão já está associado a um usuário.
 - i) Se estiver:
 - (1) Fecha a conexão (erro do cliente)
 - ii) Se não estiver:
 - (1) Testa se:
 - login tem o tamanho apropriado;
 - senha tem o tamanho apropriado; e
 - não existe usuário já cadastrado com o mesmo login.
 - (a) Se válido:
 - (i) Cadastra o usuário e associa com o socket.
 - (ii) Envia `CMD_LOGIN_OK`.
 - (b) Se não for válido:
 - (i) Envia `CMD_LOGIN_INVALIDO`.
 - (ii) Fecha a conexão.
- 2) Se for `CMD_LOGIN_USER`:
 - a) Testa se o cliente associado à conexão já está associado a um usuário.
 - i) Se estiver:
 - (1) Fecha a conexão (erro do cliente).
 - ii) Se não estiver:
 - (1) Testa se:
 - login tem o tamanho apropriado;
 - senha tem o tamanho apropriado;
 - já existe usuário cadastrado com esse login; e
 - esse usuário já não está conectado em outro socket.
 - (a) Se válido:
 - (i) Associa o usuário com o socket.
 - (ii) Envia `CMD_LOGIN_OK`.
 - (iii) Testa se existem mensagens armazenadas que:
 - sejam destinadas ao usuário; e
 - tenham status `MSG_RECEBIDA`.
 1. Caso existam, para cada mensagem:
 - a. Envia a mensagem para o cliente recém-conectado.
 - b. Muda status da mensagem para `MSG_ENTREGUE`.
 - c. Caso o remetente esteja conectado, envia `CMD_MSG_ENTREGUE` para o cliente remetente.
 - (iv) Testa se existem mensagens armazenadas que:
 - foram remetidas pelo usuário; e
 - tenham status `MSG_LIDA`.
 1. Caso existam, para cada mensagem:
 - a. Envia a confirmação de leitura para o cliente recém-conectado.
 - b. Remove a mensagem do buffer.
 - (b) Se não for válido:
 - (i) Envia `CMD_LOGIN_INVALIDO`.
 - (ii) Fecha a conexão.
 - 3) Se for `CMD_LOGIN_OK`:
 - a) Fecha a conexão (erro do cliente).
 - 4) Se for `CMD_LOGIN_INVALIDO`:
 - a) Fecha a conexão (erro do cliente).

- 5) Se for `CMD_NOVA_MSG`:
- a) Testa se:
- a Id é válida (a última Id anterior é menor que Id atual);
 - o usuário é válido (tamanho correto) e está cadastrado; e
 - o texto é válido (tamanho correto)
- i) Se válido:
- (1) Armazena a mensagem com status `MSG_RECEBIDA`.
 - (2) Envia `CMD_MSG_RECEBIDA` para o remetente.
 - (3) Se o destinatário estiver conectado:
 - (a) Envia a mensagem para o destinatário.
 - (b) Envia `CMD_MSG_ENTREGUE` para o remetente.
- ii) Se não for válido:
- (1) Envia `CMD_ID_INVALIDA`, `CMD_USER_INVALIDO` ou `CMD_MSG_INVALIDA` para o remetente, conforme o tipo de erro.
- 6) Se for `CMD_MSG_RECEBIDA`:
- a) Fecha a conexão (erro do cliente).
- 7) Se for `CMD_MSG_ENTREGUE`:
- a) Fecha a conexão (erro do cliente).
- 8) Se for `CMD_MSG_LIDA1`:
- a) Testa se:
- existe no buffer mensagem com aquela Id; e
 - o remetente dessa mensagem é o usuário enviado como parâmetro.
- i) Caso exista:
- (1) Altera o status para `MSG_LIDA`.
 - (2) Se o remetente estiver conectado, envia `CMD_MSG_LIDA2` para o remetente e remove a mensagem do buffer de mensagens.
- ii) Caso não exista:
- (1) Fecha a conexão (erro do cliente).
- 9) Se for `CMD_MSG_LIDA2`:
- a) Fecha a conexão (erro do cliente).
- 10) Se for `CMD_ID_INVALIDA`, `CMD_USER_INVALIDO` ou `CMD_MSG_INVALIDA`:
- a) Fecha a conexão (erro do cliente).
- 11) Se for `CMD_LOGOUT_USER`:
- a) Desloga o cliente e fecha a conexão.
- 12) Se for um comando desconhecido:
- a) Fecha a conexão (erro do cliente).