

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 sem sinal (`uint32_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 sem sinal de 4 bytes (`uint32_t`) contendo o número N de bytes do parâmetro
- 2.2. Em seguida, são transmitidos N bytes com o parâmetro propriamente dito.

O programa servidor, ao encerrar, deve armazenar suas informações em arquivo, de tal forma que, ao ser lançado novamente, esteja no mesmo estado anterior. Devem ser salvos e lidos ao iniciar:

- login e senha de todos os usuários cadastrados.
- todas as mensagens ainda não entregues, com seu remetente, destinatário e texto.

### COMANDOS

NOME	VALOR	PARÂMETRO 1	PARÂMETRO 2	PARÂMETRO 3
CMD_NEW_USER	1001	login: string	senha: string	-
CMD_LOGIN_USER	1002	login: string	senha: string	-
CMD_LOGIN_OK	1003	-	-	-
CMD_LOGIN_INVALIDO	1004	-	-	-

CMD_NOVA_MSG	1005	Id: uint32_t	usuário: string	texto: string
CMD_MSG_RECEBIDA	1006	Id: uint32_t	-	-
CMD_MSG_ENTREGUE	1007	Id: uint32_t	-	-
CMD_MSG_LIDA1	1008	Id: uint32_t	usuário: string	-
CMD_MSG_LIDA2	1009	Id: uint32_t	-	-
CMD_ID_INVALIDA	1010	Id: uint32_t	-	-
CMD_USER_INVALIDO	1011	Id: uint32_t	-	-
CMD_MSG_INVALIDA	1012	Id: uint32_t	-	-
CMD_LOGOUT_USER	1013	-	-	-

## CASOS DE USO

### A – Cadastro de novo usuário

- 1) O usuário digita login e senha.
- 2) O cliente se conecta ao servidor e envia CMD\_NEW\_USER (caso de uso I), com parâmetros:
  - login: string
  - senha: string
- 3) O servidor testa se o novo usuário é válido e se já não existe outro com mesmo login:
  - a) Se for válido:
    - i) Cadastra o novo usuário.
    - ii) Conecta o cliente, associando com o nome de login enviado.
    - iii) Responde CMD\_LOGIN\_OK (Caso de uso H), sem parâmetros.
    - iv) O cliente pode começar a utilizar a conexão para troca de mensagens.
  - b) Se não for válido:
    - i) Responde CMD\_LOGIN\_INVALIDO (Caso de uso H), sem parâmetros.
    - ii) Servidor fecha a conexão.

### B – Conexão de usuário

- 1) O usuário digita login e senha.
- 2) O cliente se conecta ao servidor e envia CMD\_LOGIN\_USER (caso de uso I), com parâmetros:
  - login: string
  - senha: string
- 3) O servidor testa se o usuário é válido:
  - usuário está cadastrado;
  - a senha está correta; e
  - já não existe nenhum outro cliente conectado com esse mesmo login.
  - a) Se for válido:
    - i) Conecta o cliente, associando com o nome de login enviado.
    - ii) Responde CMD\_LOGIN\_OK (Caso de uso H), sem parâmetros.
    - iii) Se for o caso, envia para o cliente todas as eventuais mensagens que sejam destinadas a ele e que estejam armazenadas com status MSG\_RECEBIDA (Caso de uso D).
    - iv) O cliente pode começar a utilizar a conexão para troca de mensagens.
  - b) Se não for válido:
    - i) Responde CMD\_LOGIN\_INVALIDO (Caso de uso H), sem parâmetros.
    - ii) Servidor fecha a conexão.

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

- 1) O usuário escolhe uma conversa (ou cria uma nova conversa, caso seja a primeira mensagem com esse outro usuário) e digita o texto de uma mensagem.
- 2) O cliente armazena a mensagem no buffer da conversa apropriada, com dados:
  - Id
  - remetente (seu próprio nome, já que o destinatário está armazenado na conversa)
  - texto
  - status MSG\_ENVIADA.
- 3) O cliente remetente envia CMD\_NOVA\_MSG para o servidor (Caso de uso I), com parâmetros:
  - Id única da mensagem: uint32\_t
  - Nome do usuário (destinatário): string
  - Texto da mensagem: string
- 4) O servidor testa se a mensagem está correta:
  - Id é única para aquele cliente.
  - Destinatário é um usuário cadastrado.
  - Texto obedece às restrições de tamanho.
- a) Se estiver correta:
  - i) Envia CMD\_MSG\_RECEBIDA para o remetente, com parâmetro Id (Caso de uso H).
  - ii) Armazena a mensagem no buffer correspondente ao remetente, com dados:
    - Id
    - destinatário
    - texto
    - status MSG\_RECEBIDA.
  - iii) Caso o destinatário esteja conectado, envia a mensagem para ele (Caso de uso D).
- b) Se não estiver correta, responde para o remetente (Caso de uso H):
  - i) Se o erro foi na Id, CMD\_ID\_INVALIDA, com parâmetro Id.
  - ii) Se o erro foi no destinatário, CMD\_USER\_INVALIDO, com parâmetro Id.
  - iii) Se o erro foi no texto, CMD\_MSG\_INVALIDA, com parâmetro Id.

#### **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 cliente conectado (continuação do caso C.4.a.iii) ou quando um cliente se conecta e existem mensagens que foram enviadas para ele por outros clientes enquanto ele esteve desconectado (continuação do caso B.3.a.iii).

- 1) O servidor seleciona uma mensagem que:
  - esteja no buffer de algum outro cliente;
  - tenha o cliente em questão como destinatário; e
  - tenha status MSG\_RECEBIDA.
- 2) O servidor envia CMD\_NOVA\_MSG para o destinatário (Casos de uso E e H), com parâmetros:
  - Id da mensagem que foi atribuída pelo remetente: uint32\_t
  - Nome do usuário (remetente): string
  - Texto da mensagem: string
- 3) O servidor altera a mensagem no buffer correspondente para o status MSG\_ENTREGUE.
- 4) O servidor envia CMD\_MSG\_ENTREGUE para o remetente, com parâmetro Id (Caso de uso H).

#### **E – Recepção de mensagem (pelo cliente destinatário)**

- 1) Caso ainda não exista uma conversa associada ao remetente, cria a nova conversa.
- 2) O cliente armazena a mensagem na conversa apropriada, com dados:

- Id
  - Remetente (o nome do outro usuário)
  - texto
  - status MSG\_ENTREGUE.
- 3) 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 cliente destinatário)

- 1) Quando uma conversa for visualizada (exibida em tela), o cliente procura todas as mensagens daquela conversa com as seguintes características:
  - O autor (remetente) é o outro cliente 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 destinatário envia para o servidor CMD\_MSG\_LIDA1 (Caso de uso I), com parâmetros:
    - Id única da mensagem: uint32\_t
    - Nome do usuário (remetente): string
  - c) O servidor altera a mensagem no buffer para o status MSG\_LIDA.
  - d) O servidor envia para o remetente CMD\_MSG\_LIDA2 (Caso de uso H), com parâmetro Id.

#### G – Desconectar usuário

- 1) O cliente envia CMD\_LOGOUT\_USER para o servidor (Caso de uso I), sem parâmetros.
- 2) O servidor desloga o cliente associado à conexão e fecha a conexão.

#### H – 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) 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 minhas 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) Muda o status da mensagem para MSG\_RECEBIDA.
    - (2) Exibe para o usuário o novo status da mensagem.
  - 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 minhas 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 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.
  - 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 minhas 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 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.
  - 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 minhas 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).

## I – 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 associado à conexão.
    - (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
        - não existe nenhum outro cliente conectado com esse mesmo login.
      - (a) Se válido:
        - (i) Cadastra o usuário associado à conexão.
        - (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 para `MSG_ENTREGUE`.
              - c. Envia `CMD_MSG_ENTREGUE` para o remetente.
        - (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 (não existe outra mensagem do mesmo remetente com mesma Id);
      - 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`:
- Fecha a conexão (erro do cliente).
- 7) Se for `CMD_MSG_ENTREGUE`:
- Fecha a conexão (erro do cliente).
- 8) Se for `CMD_MSG_LIDA1`:
- Testa se:
    - existe no buffer mensagem com aquela Id; e
    - o remetente é o usuário enviado como parâmetro.
  - Caso exista:
    - Altera o status para `MSG_LIDA`.
    - Envia `CMD_MSG_LIDA2` para o remetente.
  - Caso não exista:
    - Fecha a conexão (erro do cliente).
- 9) Se for `CMD_MSG_LIDA2`:
- Fecha a conexão (erro do cliente).
- 10) Se for `CMD_ID_INVALIDA`, `CMD_USER_INVALIDO` ou `CMD_MSG_INVALIDA`:
- Fecha a conexão (erro do cliente).
- 11) Se for `CMD_LOGOUT_USER`:
- Desloga o cliente e fecha a conexão.
- 12) Se for um comando desconhecido:
- Fecha a conexão (erro do cliente).