Aluno: João Pedro Soares da Franca.
Prontuário: PE3021114.
Sistemas Distribuídos.

Protocolo de Validação de Documentos (PVD)

1. Visão Geral

1.1. Introdução

Este documento detalha o "Protocolo de Validação de Documentos" (PVD), um protocolo de camada de aplicação projetado para a verificação de autenticidade de números de Cadastro de Pessoas Físicas (CPF) e Cadastro Nacional da Pessoa Jurídica (CNPJ) do Brasil. O protocolo foi desenvolvido para ser simples e legível, utilizando uma comunicação baseada em texto sobre TCP, o que facilita a depuração e a implementação em diversas plataformas.

1.2. Modelo de Comunicação

O PVD opera sobre um modelo cliente-servidor síncrono. O cliente inicia uma conexão TCP com o servidor, envia comandos de texto linha por linha e aguarda uma resposta correspondente do servidor para cada comando enviado. A conexão pode ou não permanecer ativa, permitindo múltiplas transações até que o cliente envie um comando para encerrá-la ou ultrapasse o limite de TimeOut.

1.3. Requisitos de Transporte

- **Protocolo:** TCP/IP
- **Porta Padrão:** 8080
- **Codificação de Caracteres:** UTF-8

2. Estrutura do Protocolo

A comunicação é baseada em mensagens de texto simples, terminadas por uma sequência de CRLF (\r\n). Cada mensagem, seja do cliente ou do servidor, representa um comando ou uma resposta completa.

2.1. Requisições do Cliente

O cliente pode enviar três tipos de comandos principais para o servidor.

2.1.1. Comando de Validação REQ

Utilizado para solicitar a validação de um número de CPF ou CNPJ.

- **Formato:** REQ <TIPO> <NUMERO>\r\n
- **Parâmetros:**
  - <TIPO>: Especifica o tipo de documento. Aceita os valores CPF ou CNPJ (não sensível a maiúsculas/minúsculas).
  - <NUMERO>: A sequência de dígitos do documento a ser validado. Formatações como pontos, hifens ou barras são removidas pelo cliente antes do envio.
- **Exemplo de Requisição CPF:** REQ CPF 12345678900\r\n
- **Exemplo de Requisição CNPJ:** REQ CNPJ 11444777000161\r\n

2.1.2. Comando de Verificação PING

Usado para verificar se o servidor está ativo e respondendo.

- **Formato:** PING\r\n

2.1.3. Comando de Encerramento QUIT

Informa ao servidor que o cliente deseja encerrar a conexão.

- **Formato:** QUIT\r\n

2.2. Respostas do Servidor

O servidor responde a cada comando do cliente com uma mensagem que consiste em um código de status numérico de 3 dígitos, seguido por uma descrição textual.

- **Formato:** <CODIGO> <MENSAGEM>\r\n

2.2.1. Respostas a Comandos REQ

Código	Mensagem de Exemplo	Descrição
200	VALID CPF 12345678900	**Sucesso:** O documento enviado é válido. A resposta inclui o tipo e o número do documento.
400	INVALID CNPJ 11444777000161	**Requisição Válida, Documento Inválido:** O documento foi processado mas considerado inválido.
422	MALFORMED	**Requisição Malformada:** A requisição não segue a sintaxe esperada (ex: REQ TIPO NUMERO).

2.2.2. Respostas a Outros Comandos

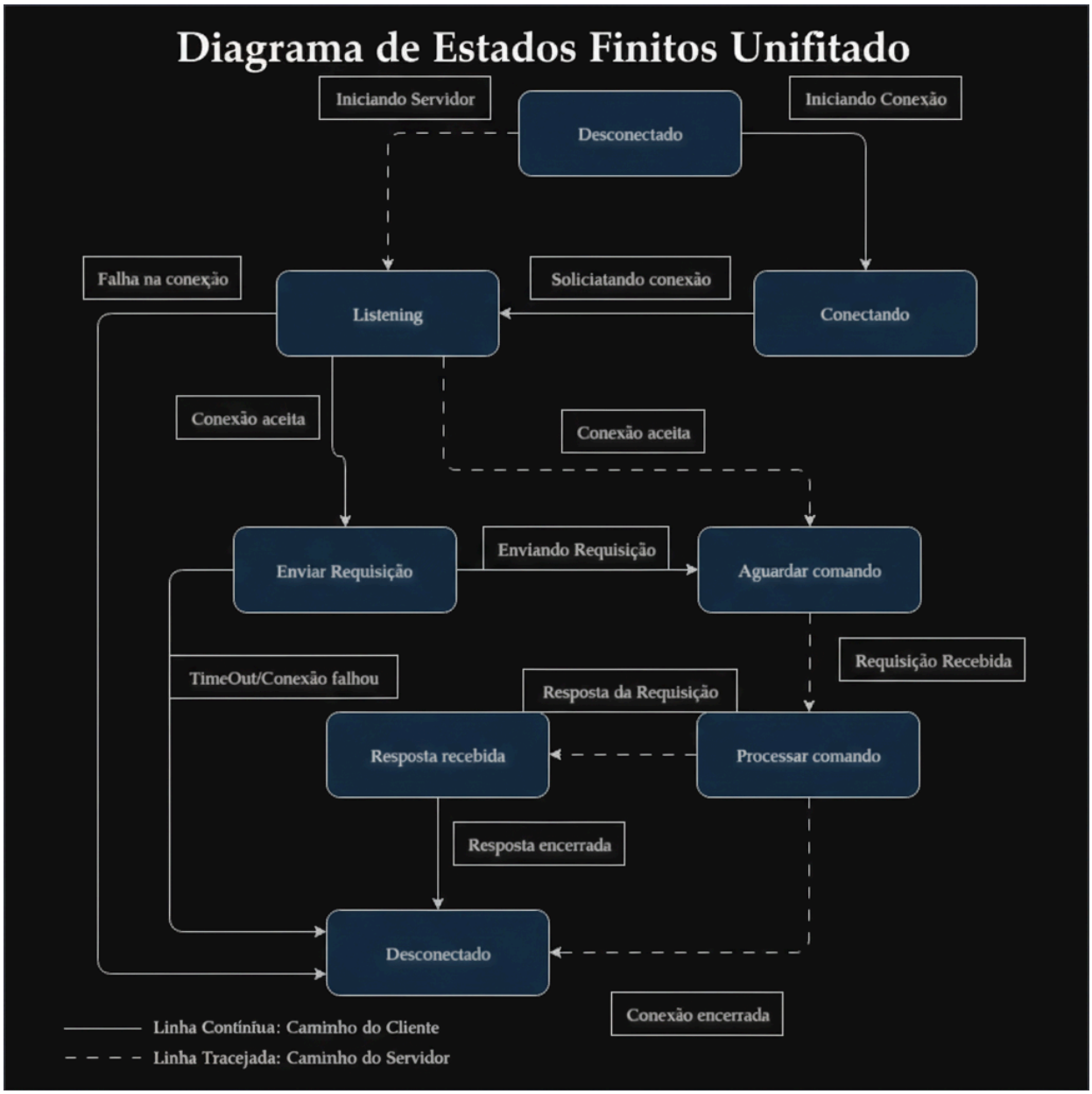
Código	Mensagem	Descrição
204	PONG	**Resposta a PING:** Confirma que o servidor está operacional.
221	BYE	**Resposta a QUIT:** Confirma o pedido de encerramento. O servidor fecha a conexão.

Código	Mensagem	Descrição
429	BUSY	**Servidor Ocupado:** O servidor atingiu o número máximo de conexões ativas.
500	ERROR	**Erro Interno:** Ocorreu um erro inesperado no servidor ao processar a requisição.

3. Diagrama de Estados Finitos Unificado

O diagrama a seguir representa a máquina de estados finitos do protocolo, unificando os ciclos de vida do cliente e do servidor em uma única visualização. Ele utiliza diferentes estilos de linha para ilustrar os caminhos percorridos por cada ponta da comunicação:

- **Linha Contínua:** Representa o caminho normal de um **Cliente**.
- **Linha Tracejada:** Representa o caminho normal de uma thread do **Servidor** ao lidar com uma conexão.



Descrição do Fluxo e dos Estados:

- **Desconectado:** O estado inicial para ambos, cliente e servidor, onde nenhuma conexão está ativa.
    - **Servidor:** Ao ser iniciado, transita para o estado Listening (linha tracejada).
    - **Cliente:** Ao iniciar uma conexão, transita para o estado Conectando (linha contínua).

- **Listening:** Um estado exclusivo do servidor. Aqui, ele aguarda passivamente por novas tentativas de conexão de clientes.
- **Conectando:** Um estado exclusivo do cliente. O cliente está ativamente tentando estabelecer uma conexão com o servidor.
    - **Sucesso:** Se a conexão for aceita por ambos, o cliente transita para Enviar Requisição e o servidor para Aguardar comando.
    - **Falha:** Em caso de falha na conexão, o cliente retorna ao estado Desconectado (como sugerido para o diagrama) ou tenta novamente, o que pode ser modelado como uma transição de volta para Conectando ou Desconectado dependendo da lógica de re-tentativa.
- **Aguardar comando:** Após aceitar uma conexão, o servidor entra neste estado, esperando que o cliente envie uma requisição.
- **Enviar Requisição:** Uma vez conectado, o cliente entra neste estado para transmitir um comando (REQ, PING, ou QUIT) ao servidor.
- **Processar comando:** Ao receber uma requisição, o servidor transita para este estado para analisar o comando, executar a validação e preparar a resposta.
- **Resposta recebida:** Após o servidor enviar a resposta, o cliente entra neste estado. Aqui, ele processa a resposta.
- **Encerramento:** A conexão pode ser encerrada de várias formas, sempre levando ao estado Desconectado:
    - O cliente encerra a conexão (após receber a resposta) e retorna a Desconectado.
    - O servidor encerra sua ponta da conexão (após processar um QUIT ou um erro) e retorna a Desconectado.
    - Ocorre um TimeOut ou falha durante a comunicação, fazendo o cliente retornar ao estado Desconectado.