
Servidor CPqD MRCP de Reconhecimento de Fala (ASR) e Síntese de Fala (TTS)

Manual de Utilização

Versão do Servidor MRCP: 1.4.0

Versão do *plugin* MRCP ASR: 2.2.0

Versão do *plugin* MRCP TTS: 1.1.0

Edição do documento: 1.2 Abril de 2017



Servidor CPqD MRCP de Reconhecimento de Fala (ASR) e Síntese de Fala (TTS)

Manual de Utilização

Versão do Servidor MRCP: 1.4.0

Versão do *plugin* MRCP ASR: 2.2.0

Versão do *plugin* MRCP TTS: 1.1.0

Edição do documento: 1.2 Abril de 2017

Número de páginas: 45

© CPqD — Todos os direitos reservados

Impresso no Brasil

As informações contidas neste documento são de propriedade do CPqD, sendo proibida a sua divulgação, reprodução ou armazenamento em base de dados ou sistema de recuperação sem permissão prévia e por escrito do CPqD. Estão sujeitas a alterações sem notificação prévia.

Os nomes de produtos, serviços ou tecnologias eventualmente mencionados neste documento são marcas registradas dos respectivos detentores.

Fazer cópias de qualquer parte deste documento para qualquer finalidade, além do uso pessoal, constitui violação das leis internacionais de direitos autorais.

Histórico de alterações

Data	Versão (SW)	Edição (doc)	Breve descrição da edição
09/02/2017	Servidor MRCP 1.0.x CPqD TTS 4.0.x	1	Versão inicial
29/03/17	Servidor MRCP 1.1.x CPqD TTS 4.0.x	1.1	Adicionado item 4.2 Alterado imagens 3.1 e 3.2 Integração com Asterisk 13
05/04/17	Servidor MRCP 1.4.x <i>plugin</i> MRCP ASR: 2.2.0 <i>plugin</i> MRCP TTS: 1.1.0	1.2	Adicionado Plugin MRCP ASR

1	INTRODUÇÃO.....	1
1.1	Objetivo.....	1
1.2	Abreviaturas.....	1
1.3	Convenções.....	1
1.4	Arquitetura do Servidor CPqD MRCP.....	2
1.5	Servidor CPqD MRCP.....	3
1.6	CPqD Texto Fala.....	3
1.7	CPqD Reconhecimento de Fala.....	3
2	INSTALAÇÃO DO SERVIDOR CPqD MRCP.....	4
2.1	Conhecimentos básicos necessários.....	4
2.2	Requisitos de instalação.....	4
	Hardware (configuração mínima):.....	4
	Software (configuração mínima).....	5
2.3	Instalação do Servidor CPqD MRCP no Linux.....	5
2.4	Execução do Servidor CPqD MRCP no Linux.....	7
2.5	Teste do Servidor CPqD MRCP no Linux.....	7
	Teste do CPqD MRCP TTS.....	7
	Teste do CPqD MRCP ASR.....	9
3	CPqD TEXTO FALA.....	11
3.1	Métodos e Eventos do MRCP para o TTS (Method).....	11
3.2	Campos do MRCP para o TTS (Header Fields).....	13
4	CPqD RECONHECIMENTO DE FALA.....	15
4.1	Métodos do MRCP para o ASR.....	15
4.2	Eventos do MRCP para o ASR.....	16
4.3	Campos do MRCP para Recog. de Fala (Header Fields).....	17
4.4	Comportamento dos temporizadores do ASR (Timers).....	18
4.5	Opções avançadas do ASR.....	18
5	CONFIGURAÇÃO DO SERVIDOR CPqD MRCP.....	19
5.1	Configuração Padrão do Servidor CPqD MRCP.....	19
5.2	Alterando os parâmetros de configuração.....	20
5.3	Recursos do Motor CPqD TTS.....	20
5.4	Recursos do Motor CPqD ASR.....	20
5.5	Parâmetros do Servidor CPqD MRCP.....	21
	Endereçamento IP.....	21
	Recursos.....	21
	Agente SIP (MRCPv2).....	22
	Transporte (MRCPv2).....	23
	Agente RTSP (MRCPv1).....	23
	Processamento de Mídia.....	24
	Protocolo de Mídia.....	24
	Plugins do Servidor.....	25
	Codec de Áudio.....	25

Histórico de edições

5.6	Parâmetros CPqD Texto Fala para MRCP.....	26
	Recursos.....	26
	Mapa de recursos.....	26
	Engine de Texto Fala.....	27
5.7	Parâmetros CPqD ASR para MRCP.....	27
	Recursos.....	27
	Mapa de recursos.....	28
	Engine de Reconhecimento de Fala.....	28
5.8	Cenários de utilização.....	29
	MRCPv1.....	29
	MRCPv2.....	29
	MRCPv1 com NAT.....	30
	MRCPv2 com NAT.....	30
5.9	Logs do Servidor CPqD MRCP.....	31
6	PLATAFORMA DE VOZ (URA).....	32
6.1	Servidor CPqD MRCP com o Avaya Voice Portal.....	32
6.2	Servidor CPqD MRCP com o Freeswitch.....	33
6.3	Servidor CPqD MRCP com o Asterisk 13.....	35
	Instalando o unimrcp-asterisk.....	35
	Configuração do MRCP no Asterisk:.....	36
	Generic Speech Recognition API:.....	36
	Unimrcp client Application:.....	37
	Exemplos de dialplan do Asterisk:.....	39

LISTA DE FIGURAS

ARQUITETURA DO SERVIDOR CPqD MRCP.....	2
APLICATIVO AUDACITY.....	8
MÁQUINA DE ESTADOS DO MRCP PARA SÍNTESE DE FALA.....	12
MÁQUINA DE ESTADOS DO MRCP ASR.....	16
CENÁRIO DE UTILIZAÇÃO CLIENTE(S) MRCPv1.....	29
CENÁRIO DE UTILIZAÇÃO CLIENTE(S) MRCPv2.....	29

LISTA DE TABELAS

CONVENÇÕES DESTE MANUAL.....	1
EXEMPLO DE INSTALAÇÃO DO SERVIDOR MRCP TTS.....	5
EXEMPLO DE INSTALAÇÃO DO SERVIDOR MRCP ASR.....	6
ESTRUTURA DO DIRETÓRIO DE INSTALAÇÃO.....	6
MÉTODOS E EVENTOS DO MRCP PARA TTS.....	11
CAMPOS DE CABEÇALHO DO MRCP PARA SÍNTESE DE FALA.....	13
MÉTODOS E EVENTOS DO MRCP PARA ASR.....	15



EVENTOS DO MRCP PARA ASR.....	16
CAMPOS DE CABEÇALHO DO MRCP PARA ASR.....	17
TEMPORIZADORES DO ASR.....	18

1 INTRODUÇÃO

1.1 Objetivo

Este documento tem por objetivo descrever a arquitetura, o procedimento de utilização, instalação e o funcionamento do Servidor CPqD MRCP em conjunto com o CPqD Texto Fala (do inglês Text-To-Speech “TTS”) e do CPqD Reconhecimento de Fala (do inglês Automatic Speech Recognition “ASR”).

1.2 Abreviaturas

- TTS — Síntese de fala, do inglês: Text-To-Speech
- ASR – Reconhecimento automático de fala, do inglês: Automatic Speech Recognition
- MRCP — Media Resource Control Protocol
- NAT – Network Address Translation
- Barge-in – Evento de interrupção do sintetizador de fala (intromissão)
- TCP – Transmission Control Protocol (Protocolo da internet)
- IP Addr – Endereçamento do computador/servidor
- RTSP – Real-Time Streaming Protocol (relacionado ao MRCPv1)
- SIP – Session Initiation Protocol (relacionado ao MRCPv2)
- RTP – Real-Time Transport Protocol (responsável pelo despacho do áudio/media)
- URA – Unidade de Resposta Audível (do inglês: “IVR” Interactive Voice Response)

1.3 Convenções

CONVENÇÃO	SIGNIFICADO
Texto em Courier New	Comandos de linha
Figuras com blocos e/ou textos sem destaque	Blocos opcionais do sistema

Tabela 1.1: Convenções deste manual

1.4 Arquitetura do Servidor CPqD MRCP

O Servidor CPqD MRCP fornece conectividade para clientes MRCP através dos protocolos específicos a versão MRCPv1 ou MRCPv2. Desta forma, o ASR e TTS do CPqD podem ser integrados de forma ágil em *Call Center* ou em *Contact Center*, além de permitir uma maior flexibilidade no desenvolvimento de diálogos presente em URA.

O MRCPv1 utiliza o protocolo de controle RTSP (Real Time Streaming Protocol) para estabelecer conexões (sessões) nas solicitações síntese e reconhecimento, enquanto o MRCPv2 conta com o protocolo SIP (Session Initiation Protocol) e com SDP (Session Description Protocol) para trocar parâmetros associadas a um diálogo. Sendo o SIP, responsável pelo estabelecimento e a interoperabilidade do diálogo.

Para realizar o transporte de áudio nas sessões (Media Session), é designado o protocolo RTP (Real Time Transport Protocol), na taxa de 8KHz (oito kilo hertz), codificação PCMA ou PCMU (8 bits por amostra).

Empresas como: Aspect, Avaya, Cisco, Dialogic e Khomp possuem implementações de clientes MRCPv1 e/ou MRCPv2. Existem também projetos de código livre que fazem utilização dos protocolos MRCP, como: Freeswitch, Asterisk e UniMRCP. A figura 1.1 ilustra a arquitetura do Servidor CPqD MRCP:

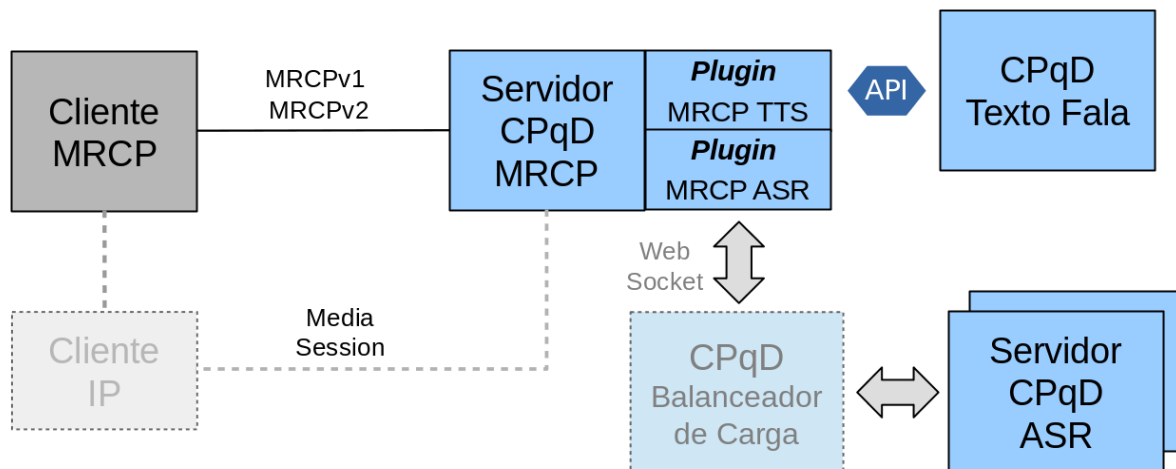


Figura 1.1: Arquitetura do Servidor CPqD MRCP

- **Cliente MRCP:** Normalmente esse componente está situado na URA ou no Portal de Voz que deseja utilizar o recurso de síntese e/ou reconhecimento de fala.
- **Servidor CPqD MRCP:** Componente servidor para prover os recursos de síntese e reconhecimento de fala.
- **Plugin MRCP TTS:** Modulo responsável por integrar o motor CPqD TextoFala com o Servidor MRCP.
- **CPqD Texto Fala:** Mecanismo CPqD TextoFala.
- **Plugin MRCP ASR:** Modulo responsável por integrar o Servidor CPqD ASR com o Servidor MRCP.
- **CPqD Balanceador de Carga:** Modulo opcional do servidor CPqD ASR, responsável pela distribuição de carga para o cenário(s) com mais de um Servidor CPqD ASR.
- **Servidor CPqD ASR:** Elemento servidor do reconhecimento de fala.

1.5 Servidor CPqD MRCP

O Servidor CPqD MRCP provê serviços de mídia baseado no protocolo MRCP (Media Resource Control Protocol) versão 1 (um) e 2 (dois) para síntese e reconhecimento de de fala.

Informações detalhadas sobre as funcionalidades e características do protocolo MRCP estão disponíveis publicamente e podem ser obtidas via internet:

- RFC4463: <http://tools.ietf.org/html/rfc4463>
- RFC6787: <https://tools.ietf.org/html/rfc6787>

1.6 CPqD Texto Fala

O CPqD Texto Fala é um framework de conversão de texto em fala desenvolvido pelo CPqD. Pode-se trabalhar com diversos tipos de texto, desde sentenças simples até grandes conjuntos de dados. Os formatos de texto suportados pelo Servidor CPqD MRCP são:

- Texto com markup SSML (`application/ssml+xml`)
- Texto puro (`text/plain`)

1.7 CPqD Reconhecimento de Fala

O CPqD Reconhecimento de fala é um mecanismo de conversão fala para texto desenvolvido pelo CPqD. O ASR permite trabalhar com reconhecimento de fala livre e gramatical com interpretação semântica.

O formato gramatical ou de fala livre suportado pelo Servidor CPqD MRCP é:

- Uniform Resource Identifier (`text/uri-list`)

2 INSTALAÇÃO DO SERVIDOR CPqD MRCP

O Servidor CPqD MRCP é composto por um conjunto de bibliotecas compiladas para o sistema operacional alvo, arquivos de configuração e ferramentas para testes e validação.

***** Atenção *****

Antes de iniciar a instalação do Servidor CPqD MRCP, certifique-se quais recursos foram adquiridos com a equipe de vendas/comercial do CPqD:

- Se CPqD Texto Fala: Tenha certeza que o mesmo esteja instalado, licenciado e validado conforme as instruções do manual:
 - “Manual_CPqDTextoFala_Instalacao.pdf”
- Se CPqD Reconhecimento de Fala, Tenha certeza que o mesmo esteja instalado, licenciado e validado conforme as instruções do manual:
 - “Servidor de Reconhecimento de Fala – Manual de Instalação.pdf”

2.1 Conhecimentos básicos necessários

Para a compreensão e execução das tarefas descritas neste manual é necessário que o usuário tenha um perfil de administrador do sistema alvo e possua conhecimentos em Linux:

- CentOS ou Red Hat Enterprise Linux
- Familiaridade com Bash.

2.2 Requisitos de instalação

A demanda do Servidor CPqD MRCP por hardware, em relação ao poder de processamento, é baixo. Para um bom desempenho, é necessário garantir uma rede de dados estável entre o Servidor e o cliente MRCP.

Hardware (configuração mínima):

- 4 GB de memória RAM
- 10 GB de HD
- 2 vCPU's @ 2.00 GHz
- Interface de rede 100 Mbps

Software (configuração mínima)

Servidor MRCP:

- Red Hat Enterprise Linux 7.2 ou CentOS 7.2, 64 bits
- Bash
- GNU libc 2.17⁽¹⁾ e libstdc++ 3.4.19⁽²⁾

*** Observações ***

1) Para descobrir a versão da GNU libc instalada no sistema, execute `ldd --version` no terminal. Normalmente, instalar o gcc versão 4.8.5 ou superior é suficiente para atender essa necessidade.

2) Para descobrir a versão da libstdc++ instalado no sistema, execute `strings $ (/sbin/ldconfig -p | grep stdc++) | grep LIBCXX` no terminal. Normalmente, instalar o g++ versão 4.8.5 ou superior é suficiente para atender esse requisito.

2.3 Instalação do Servidor CPqD MRCP no Linux

1. Se existir qualquer instalação do CPqD MRCP TTS ou ASR, o mesmo deve ser apagado antes de continuar essa instalação:

```
$ rm -r /opt/cpqd/mrcp/
```

2. Execute o(s) arquivo(s) instalador(es) conforme adquirido com a equipe de vendas/comercial do CPqD:

se TTS: *cpqd-tts-mrcp-server-OS-VERSION.run*

se ASR: *cpqd-asr-mrcp-server-OS-VERSION.run*

se ambos: Necessário executar os dois instaladores e siga as instruções que aparecerão no terminal (veja as tabelas 2.1 e 2.2)

Todo o sistema será instalado abaixo do diretório `/opt/cpqd/mrcp` e os logs serão criados abaixo de `/opt/cpqd/mrcp/server/log`, veja o item 5.9. Certifique-se de o espaço em disco definido anteriormente esteja disponível para este diretório.

```
$ ./cpqd-tts-mrcp-server-OS-X.X.X.run
Verifying archive integrity... 100% All good.
Uncompressing cpqd-tts-mrcp-OS-X.X.X 100%

Enter the IP address that the MRCP Server will use to receive requests:
192.168.10.1
Inform the installed path of CPqD TTS: /opt/cpqd/tts/
Configuration done.
All done.
```

Tabela 2.1: Exemplo de instalação do Servidor MRCP TTS

- O script de instalação do CPqD TTS MRCP SERVER solicitará duas informações:
 - “Informe o local de instalação do CPqD Texto Fala”: aqui deve ser informado o diretório de instalação do CPqD Texto Fala, **por padrão: /opt/cpqd/tts/**
 - “Digite o endereço IP pelo qual o Servidor MRCP receberá requisições”: aqui deve ser informado o endereço IP do Servidor MRCP que trocará mensagens com o cliente.
Nota: O IP informado não pode ser 127.0.0.1 ou 0.0.0.0.

```

$ ./cpqd-asr-mrcp-server-OS-X.Y.Z.run
Verifying archive integrity... 100% All good.
Uncompressing cpqd-asr-mrcp-server-OS-Z.Y.Z 100%

MRCP Server already installed!
Installing CPqD MRCP ASR...
Inform the CPqD ASR Websocket Server IP address: 192.168.10.1
Configuration done.
All done.
    
```

Tabela 2.2: Exemplo de instalação do Servidor MRCP ASR

- O script de instalação do CPqD ASR MRCP SERVER solicitará duas informações:
 - “Informe o endereço IP do Servidor Websocket ASR”: aqui deve ser informado o endereço IP da máquina em que o servidor de reconhecimento de fala do CPqD está instalado.
Nota: O IP informado não pode ser 127.0.0.1 ou 0.0.0.0.
 - “Digite o endereço IP pelo qual o Servidor MRCP receberá requisições”: aqui deve ser informado o endereço IP da máquina em o Servidor MRCP está instalado.
Nota: Se não foi solicitado o endereço IP, o mesmo foi informado na instalação do MRCP TTS (ou vice e versa)

3. Ao final do processo, a estrutura de diretórios deve ser similar àquela mostrada na Tabela 2.3.

```

/opt/cpqd/mrcp/
├── README-ASR.md
├── README-TTS.md
├── server
│   ├── bin
│   ├── conf
│   ├── data
│   ├── lib
│   ├── log
│   ├── plugin
│   └── var
    
```

Tabela 2.3: Estrutura do diretório de instalação.

2.4 Execução do Servidor CPqD MRCP no Linux

O Servidor CPqD MRCP pode ser executado com o seguinte comando:

```
$ /opt/cpqd/mrcp/server/bin/start-server --run background
```

Os logs serão gravados em: (XX representa numero crescente do log)

```
$ ls /opt/cpqd/mrcp/server/unimrcpserver-XX.log
```

Para checar se o servidor está rodando execute o comando:

```
$ /opt/cpqd/mrcp/server/bin/server-status
```

Para parar o servidor, execute o comando:

```
$ /opt/cpqd/mrcp/server/bin/stop-server
```

2.5 Teste do Servidor CPqD MRCP no Linux

A ferramenta “mrcp_client”, instalada juntamente com o Servidor CPqD MRCP permite realizar síntese de fala e reconhecimento de fala para validar a instalação que foi feita. Veja o passo a passo de utilização da ferramenta.(A ferramenta não valida configurações de *Firewall*)

Teste do CPqD MRCP TTS

1. Comando para iniciar o cliente MRCP:

```
$ /opt/cpqd/mrcp/server/bin/mrcp_client
```

2. Comando para testar TTS via MRCPv1:

```
$ > run synth uni1
```

Texto esperado:

```
Session: ecf609348e7a4889  
Content-Type: application/mrcp  
Content-Length: 68  
SPEAK-COMplete 1 COMPLETE MRCP/1.0  
Completion-Cause: 000 normal
```

Áudio esperado:

```
/opt/cpqd/mrcp/server/var/synth-8kHz-ecf609348e7a4889.pcm
```

3. Comando para testar TTS via MRCPv2:

```
$ > run synth uni2
```

Texto esperado:

```
MRCP/2.0 122 SPEAK-COMplete 1 COMPLETE
Channel-Identifier: b688a27fb4fa436a@speechsynth
Completion-Cause: 000 normal
```

Áudio esperado:

```
/opt/cpqd/mrcp/server/var/synth-8kHz-b688a27fb4fa436a.pcm
```

Se a instalação não estiver correta, erros serão lançados no terminal e deve ser encaminhado o(s) arquivo(s) de auditoria (logs) para o CPqD analisar a(s) inconsistência(s).

4. Para sair do cliente MRCP:

```
$ > quit (ou CrtL C)
```

5. Para reproduzir o arquivo de áudio sintetizado, copie o(s) arquivo(s) para uma máquina Linux local que tenha caixas de som ou fones de ouvido conectado :

```
$ sudo yum install sox | play -t raw -r 8000 -c 1 -b 16 -e signed-integer synth-8kHz-b688a27fb4fa436a.pcm
```

6. Outra opção para validar a síntese: Instalar o aplicativo "**Audacity**".

Instruções (Windows ou Linux):

- 1) Clicar em: Ficheiro->Importar->Dados Raw ...
- 2) Selecionar o arquivo: Localização = synth-8kHz-XXXXXXXXX.pcm
- 3) Selecionar as opções na sequência: Codificação = Signed 16 bit PCM; Sem endianness; 1 canal; 0 bytes; 100 % e 8000 Hz.
- 4) Para escutar, pressionar a tecla "ESPAÇO" ou clicar "Play"

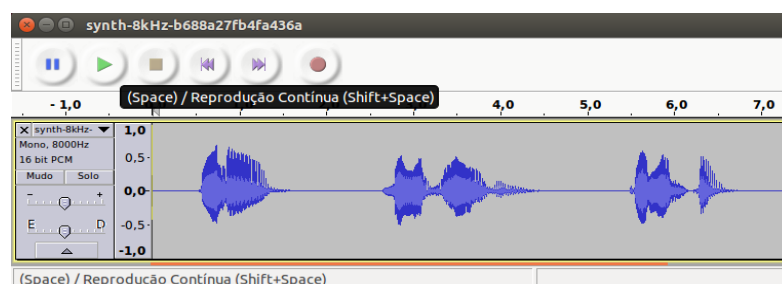


Figura 2.1: Aplicativo Audacity

Teste do CPqD MRCP ASR

1. Comando para iniciar o cliente MRCP:
\$ /opt/cpqd/mrcp/server/bin/mrcp_client
2. Comando para testar o Reconhecimento de Fala Livre via MRCPv2:
\$ > run recog-slm uni2

Texto esperado:

agora ele despenca do teto em direção ao piso

3. Comando para testar reconhecimento gramatical via MRCPv1:
\$ > run recog-grammar uni1

Texto esperado:

eu quero uma pizza vegetariana

Se a instalação não estiver correta, erros serão lançados no terminal e deve ser encaminhado os arquivos de auditoria (logs) para o CPqD analisar a(s) inconsistência(s).

4. Para sair do cliente MRCP:
\$ > quit (ou CrtL C)

3 CPqD TEXTO FALA

3.1 Métodos e Eventos do MRCP para o TTS (Method)

Conforme descrito nas especificações MRCPv1 e MRCPv2, o Servidor CPqD MRCP para síntese fala (TTS) suportam os métodos e eventos, como descrito na tabela 3.1:

Métodos / Eventos	MRCPv1	MRCPv2	Servidor CPqD	Descrição
SPEAK	X	X	X	Inicializa a síntese de fala e fornece o <i>streamming</i> de áudio
PAUSE	X	X	X	Pausa a síntese de fala solicitado pelo SPEAK
RESUME	X	X	X	Retoma a síntese que foi pausada pelo PAUSE
STOP	X	X	X	Interrompe a síntese solicitado pelo SPEAK
BARGE-IN-OC.	X	X	X	Sinaliza o evento Barge-In para o sintetizador
CONTROL	X	X	-	Modifica a ação da síntese que está em percurso
DEFINE-LEX.	-	X	-	Define o léxico da pronúncia do sintetizador
SET-PARAMS	X	X	X	Parametriza variáveis do sintetizador de fala
GET-PARAMS	X	X	X	Informa valor das variáveis do sintetizador
SPEAK-COMPLETE	X	X	X	Ocorre quando o pedido SPEAK foi finalizado
SPEECH-MARKER	X	X	-	Sinalizado quando o sintetizador encontra o elemento <mark>.

Tabela 3.1: Métodos e Eventos do MRCP para TTS

O recurso de síntese de fala segue uma máquina de estados comandada pelas solicitações do cliente MRCPv1/MRCPv2 ou pelos eventos do próprio sintetizador, veja o diagrama 3.1 :

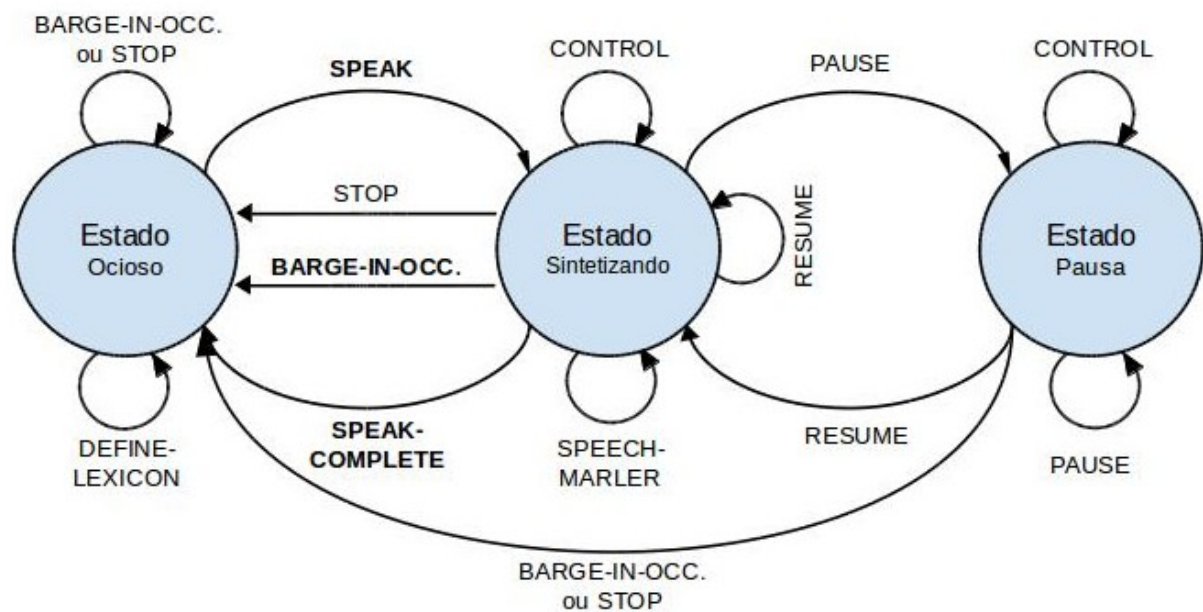


Figura 3.1: Máquina de estados do MRCP para Síntese de Fala

3.2 Campos do MRCP para o TTS (Header Fields)

A tabela 3.2 descreve os campos de cabeçalho presente nos métodos e eventos do MRCPv1 ou MRCPv2 para o recurso de TTS. Os métodos estão divididos em dois grupos, “Requisição” significa que a mensagem será trocada no sentido do Cliente para o Servidor, e “Resposta” no sentido oposto.

Nome Campo	MRCPv1	MRCPv2	Servidor CPqD MRCP	Metodos		Eventos	SET GET PARAMS
				Requisição	Resposta		
Completion-Cause	X	X	X	-	-	SPEAK-COMplete	Não
Completion-Reason	-	X	-	-	-	SPEAK-COMplete	Não
Failed-URI	X	X	-	-	SPEAK *DEFINE-LEX.	SPEAK-COMplete	Não
Failed-URI-Cause	X	X	-	-	SPEAK BARGE-IN-OC. *DEFINE-LEX.	SPEAK-COMplete	Não
Speech-Marker	X	X	-	-	SPEAK BARGE-IN-OC. STOP *CONTROL	*SPEECH-MARKER	Não
Voice-Gender	X	X	-				
Voice-Age	X	X	-	SPEAK	-	-	Sim
Voice-Variant	X	X	-	*CONTROL			
Voice-Name	X	X	X				
Prosody-Pitch	X	X	-				
Prosody-Contour	X	X	-				
Prosody-Range	X	X	-	SPEAK	-	-	Sim
Prosody-Rate	X	X	X	*CONTROL			
Prosody-Duration	X	X	-				
Prosody-Volume	X	X	X				
Speaker-Profile	X	X	-	SPEAK	-	-	Sim
Speech-Language	X	X	-	SPEAK	-	-	Sim
Kill-On-Barge-In	X	X	X	SPEAK	-	-	Sim
Fetch-Hint	X	X	-	SPEAK	-	-	Sim
Audio-Fetch-Hint	X	X	-	SPEAK	-	-	Sim
Jump-Target / Jump-Size	X	X	-	SPEAK *CONTROL	-	-	Não
Speak-Restart	X	X	-	-	*CONTROL	-	Não
Speak-Length	X	X	-	SPEAK *CONTROL	-	-	Não
Speak-Length	X	X	-	*DEFINE-LEXICON	-	-	Não
Load-Lexicon	-	X	-	SPEAK	-	-	Sim
Lexicon-Search-Order	-	X	-	SPEAK	-	-	Sim

Tabela 3.2: Campos de cabeçalho do MRCP para Síntese de Fala

4 CPqD RECONHECIMENTO DE FALA

4.1 Métodos do MRCP para o ASR

Conforme descrito nas especificações MRCPv1 e MRCPv2, o Servidor CPqD MRCP para reconhecimento de fala (ASR) suportam os métodos , como descrito na tabela 4.1 :

Métodos	MRCPv1	MRCPv2	Servidor CPqD	Descrição
SET-PARAMS	X	X	X	Parametriza variáveis do reconhecimento de fala
GET-PARAMS	X	X	X	Informa valor das variáveis do reconhecimento de fala
DEFINE-GRAMMAR	X	X	X	Fornece a gramática para reconhecimento de fala
RECOGNIZE	X	X	X	Inicia o reconhecimento de fala atrelada a gramática.
INTERPRET	-	X	-	Solicita que o reconhecimento de fala faça uma interpretação.
GET-RESULT	X	X	-	Recupera o resultado do reconhecimento.
REC.-START-TIMERS	X	-	X	Inicia os temporizadores do reconhecedor (MRCPv1).
START-INPUT-TIMERS	-	X	X	Inicia os temporizadores do reconhecedor (MRCPv1).
STOP	X	X	-	Cancela o reconhecimento ativo.

Tabela 4.1: Métodos e Eventos do MRCP para ASR

O recurso de reconhecimento de fala segue uma máquina de estados comandada pelas solicitações do cliente MRCPv1/MRCPv2 ou pelos eventos do próprio reconhecedor, veja o diagrama abaixo 4.1:

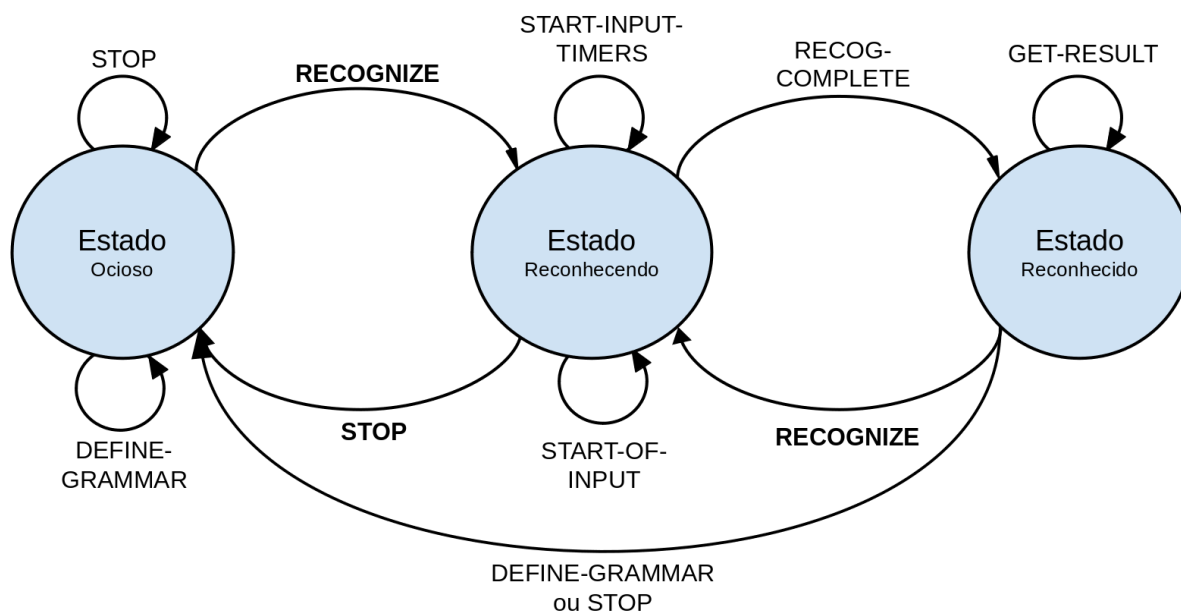


Figura 4.1: Máquina de estados do MRCP ASR

4.2 Eventos do MRCP para o ASR

Os eventos do ASR estão descritos na tabela abaixo :

Eventos	MRCPv1	MRCPv2	Servidor CPqD	Descrição
START-OF-INPUT	-	X	X	Ocorre quando o reconhecedor detecta início de fala.
START-OF-SPEECH	X	-	X	Ocorre quando o reconhecedor detecta início de fala.
RECOG.-COMPLETE	X	X	X	Sinaliza que o reconhecimento foi finalizado.
INTERP.-COMPLETE	X	X	-	Sinaliza que o interpretador semântico foi finalizado.

Tabela 4.2: Eventos do MRCP para ASR

4.3 Campos do MRCP para Recog. de Fala (Header Fields)

A tabela XX descreve os campos de cabeçalho presente nos métodos e eventos do MRCPv1 ou MRCPv2 para o recurso de ASR. Os métodos estão divididos em dois grupos, “Requisição” significa que a mensagem será trocada no sentido do Cliente para o Servidor, e “Resposta” no sentido oposto.

Nome do Campo (Header)	MRCPv1	MRCPv2	Servidor CpqD	Metodos		Eventos	SET GET PARAMS
				Requisição	Resposta		
Completion-cause	X	X	X	-	RECOGNIZE, DEFINE-GRAM	RECOG.- COMPLETE	NÃO
Completion-Reason	-	X	-	-	RECOGNIZE, DEFINE-GRAM	RECOG.- COMPLETE	NÃO
Failed-URI	X	X	-	-	RECOGNIZE, DEFINE-GRAM	RECOG.- COMPLETE	NÃO
Failed-URI-Cause	X	X	-	-	RECOGNIZE, DEFINE-GRAM	RECOG.- COMPLETE	NÃO
Recognition-Mode		X	-	RECOGNIZE	-	-	NÃO
Input-Type	-	X	-	-	-	START-OF-INPUT	NÃO
Confidence-Threshold	X	X	X	RECOGNIZE, GET-RESULT	-	-	SIM
Sensitivity-Level	X	X	-	RECOGNIZE	-	-	SIM
Speed-vc-Accuracy	X	X	-	RECOGNIZE	-	-	SIM
N-Best-List-Length	X	X	X	RECOGNIZE, GET-RESULT	-	-	SIM
No-Input-Timeout	X	X	X	RECOGNIZE	-	-	SIM
Recognition-Timeout	X	X	X	RECOGNIZE	-	-	SIM
Speech-Complete-Timeout	X	X	-	RECOGNIZE	-	-	SIM
Speech-Incomplete-Timeout	X	X	-	RECOGNIZE	-	-	SIM
Hotword-Max-Duration	-	X	-	RECOGNIZE	-	-	SIM
Hotword-Min-Duration	-	X	-	RECOGNIZE	-	-	SIM
DTMF-Interdigit-Timeout	X	X	-	RECOGNIZE	-	-	SIM
DTMF-Term-Timeout	X	X	-	RECOGNIZE	-	-	SIM
DTMF-Term-Char	X	X	-	RECOGNIZE	-	-	SIM
DTMF-Buffer-Time	-	X	-	-	-	-	SIM
Clear-DTMF-Buffer	-	X	-	RECOGNIZE	-	-	NÃO
Save-Waveform	X	X	-	RECOGNIZE	-	-	SIM
Waveform-URI	-	X	-	-	-	RECOG.- COMPLETE	NÃO
Waveform-URL	X	-	-	-	-	RECOG.- COMPLETE	NÃO
Input-Waveform-URI	-	X	-	RECOGNIZE	-	-	NÃO
Media-Type	-	X	-	RECOGNIZE	-	-	SIM
Start-Input-Timers	-	X	X	RECOGNIZE	-	-	NÃO
Recognizer-Start-Timers	X	-	X	RECOGNIZE	-	-	NÃO
Speech-Language	X	X	-	RECOGNIZE, DEFINE-GRAM	-	-	SIM
Cancel-If-Queue		X	-	RECOGNIZE	-	-	NÃO
New-Audio-Channel	X	X	-	RECOGNIZE	-	-	NÃO
Ver-Buffer-Utterance	-	X	-	RECOGNIZE	-	-	NÃO
Early-No-Match	-	X	-	RECOGNIZE	-	-	SIM
Interpret-Text	-	X	-	INTERPRET	-	-	NÃO
Recognizer-Context-Block	X	X	-	-	-	-	SIM
Fetch-Timeout	X	-	-	RECOGNIZE, DEFINE-GRAM	-	-	SIM

Tabela 4.3: Campos de cabeçalho do MRCP para ASR

4.4 Comportamento dos temporizadores do ASR (Timers)

O protocolo MRCPv1 e MRCPv2 define alguns temporizadores (*timers*) com o propósito de controlar as atividades e comportamentos do reconhecimento. Veja uma descrição resumida conforme a tabela abaixo:

Timer Name	Propósito	Quando inicializa o temporizador?	Quando expira? (Completion-Cause)
No-Input-Timeout	Para o caso que o usuário não responda.	Através dos métodos: RECOGNIZE (se start-input-timer=true) ou START-INPUT-TIMERS	002 – no-input-timeout
Recognition-Timeout	Limita o tempo para reconhecer a fala.	Via o método: START-OF-INPUT	Se MRCPv1: 003 – recognition-timeout Se MRCPv2: 015-no-match-maxtime

Tabela 4.4: Temporizadores do ASR

4.5 Opções avançadas do ASR

O mecanismo de reconhecimento de fala do CPqD possui configurações avançadas com o propósito de oferecer uma melhor experiência (usabilidade) ao usuário que irá utilizar o reconhecimento de fala.

- **Endpoint:** Responsável por detectar as bordas de início e final de fala do sinal.
- **TuningLog:** Funcionalidade para auxiliar os ajustes de melhoramento do ASR.
- **Decoder:** Recursos avançados para aperfeiçoamento das aplicações de URA (*nbest*, *buffer*, modelos acústicos, modelos linguísticos, etc...).

Para acessar ou reconfigurar estes recursos/parâmetros, entre em contato com o CPqD para receber todo o suporte necessário e orientações técnicas conforme o caso.

5 CONFIGURAÇÃO DO SERVIDOR CPqD MRCP

Este capítulo descreve a configuração padrão do Servidor, os principais parâmetros e exemplos de configuração dos cenários mais comuns. Os arquivos de configuração do Servidor CPqD MRCP utilizam o formato de configuração XML (Extended Markup Language). Desejável conhecimento básico de XML para editar o arquivo de configuração do servidor.

5.1 Configuração Padrão do Servidor CPqD MRCP

O Servidor MRCP CPqD possui uma configuração padrão, conforme descrito:

MRCPv1:

- porta RTSP: 1554
- faixa de porta RTP: 5000 a 6000
- resource path: speechsynthesizer

MRCPv2:

- porta SIP: 8060
- porta MRCPv2 Agent: 1544
- faixa de porta RTP: 5000 a 6000

Codecs de áudio suportados:

- PCMA (8kHz)
- PCMU (8kHz)

Nível do LOG:

- INFO (mensagens informativas)

5.2 Alterando os parâmetros de configuração

Para alterar o ip e portas do servidor edite o arquivo:

- /opt/cpqd/mrcp/server/conf/templates/unimrcpserver.xml

Para efeito, após edição do arquivo, execute o comando, com o servidor desligado:

```
$ /opt/cpqd/mrcp/server/bin/configure
```

Para maiores detalhes dos parâmetros veja **item 5.5**.

***** Atenção *****

**Nunca editar arquivo abaixo, ele é gerado automaticamente:
/opt/cpqd/mrcp/server/conf/unimrcpserver.xml**

Para alterar parâmetros específicos da instalação do CPqD TTS, edite o arquivo:

- /opt/cpqd/mrcp/server/conf/cpqd/cpqd-tts.xml

Para alterar parâmetros específicos da instalação do CPqD ASR, edite o arquivo:

- /opt/cpqd/mrcp/server/conf/cpqd/cpqd-asr.xml

Para efeito, após edição do arquivo, execute o comando, com o servidor desligado:

```
$ /opt/cpqd/mrcp/server/bin/configure
```

5.3 Recursos do Motor CPqD TTS

Serão aplicados os valores definidos por padrão a todas as sessões de síntese de fala, a menos que o cliente MRCP especifique outros valores via métodos e campos do MRCP, como descrito na tabela **3.2**. Os valores pré definidos e parâmetros suportados pelo CPqD Texto Fala estão detalhados nos documentos:

- [Manual CPqDTextoFala Manual Formatacao.pdf](#)
- [Manual CPqDTextoFala SSML.pdf](#)

5.4 Recursos do Motor CPqD ASR

Serão aplicados os valores definidos por padrão a todas as sessões de reconhecimento de fala, a menos que o cliente MRCP especifique outros valores via métodos e campos do MRCP, como descrito na tabela **4.3**. Veja as instruções de como desenvolver e utilizar gramáticas em nosso servidor:

- "manual_desenvolvimento_gramaticas.pdf"
- "manual_uso_gramaticas.pdf"

5.5 Parâmetros do Servidor CPqD MRCP

Os parâmetros de configuração do Servidor CPqD MRCP são ajustados por meio da atribuição de valores a variáveis pré definidas. Qualquer alteração será válido somente após o reinicialização do Servidor.

Os parametros abaixo podem ser alterados no arquivo:

- /opt/cpqd/mrcp/server/conf/templates/unimrcpserver.xml

Endereçamento IP

Parâmetros	Descrição
<ip>	Especifica o endereço IP local.
<ext-ip>	Especifica o endereço IP sobre o NAT externo.
Código	<pre><ip>10.10.0.1</ip> <ext-ip>10.10.0.1</ext-ip> <!-- <ip type="auto"/> --> <!-- <ip type="iface">eth0</ip>--></pre>

Atributos	Descrição
type	Define o tipo de endereço IP: numérico, auto ou interface.

Recursos

Parâmetro	Descrição
<resource-factory>	Define os recursos que serão carregados
Código	<pre><resource-factory> __MRCP_RESOURCE_FACTORY__ </resource-factory></pre>

Atributos	Descrição
__MRCP_RESOURCE_FACTORY__	Campo preenchido automaticamente pelo <i>configure</i> . Não editar. Veja item 4.5.

Agente SIP (MRCPv2)

Parâmetros	Descrição
<sip-uas>	Especifica o agente SIP para o MRCPv2
Código	<pre><sip-uas id="SIP-Agent-1" type="SofiaSIP"> <!-- <sip-ip> </sip-ip> --> <!-- <sip-ext-ip> </sip-ext-ip> --> <sip-port>8060</sip-port> <sip-transport>udp,tcp</sip-transport> <ua-name>UniMRCP SofiaSIP</ua-name> <sdp-origin>UniMRCPServer</sdp-origin> </sip-uas></pre>

Atributos	Descrição
<sip-ip>	Especifica o endereço IP (agente SIP), se não especificar, será utilizado o valor do parâmetro <ip>
<sip-ext-ip>	Especifica o IP sobre NAT (agente SIP), se não especificar, assume o valor do parâmetro <ext-ip>
<sip-port>	Especifica a porta local do agente SIP
<sip-transport>	Especifica modos de transporte TCP ou UDP
<ua-name>	Especifica o nome do agente trafegado nas mensagens SIP
<sdp-origin>	Especifica o campo SDP de origem

Transporte (MRCPv2)

Parâmetros	Descrição
<mrcpv2-uas>	Especifica o agente MRCPv2 responsável pelo transporte.
Código	<pre><mrcpv2-uas id="MRCPv2-Agent-1"> <mrcp-port>1544</mrcp-port> <max-connection-count>100</max-connection-count> <force-new-connection>>false</force-new-connection> </mrcpv2-uas></pre>

Atributos	Descrição
<mrcp-ip>	Especifica o endereço IP do agente de transporte MRCPv2, por padrão, mesmo valor de <ip>
<mrcp-port>	Especifica a porta local do agente MRCPv2
<max-connection-count>	Especifica o número máximo de conexões TCP/MRCPv2 por agente.
<force-new-connection>	Especifica que o cliente deverá forçar novas conexões.

Agente RTSP (MRCPv1)

Parâmetros	Descrição
<rtsp-uas>	Especifica o agente RTSP para o MRCPv1
Código	<pre><rtsp-uas id="RTSP-Agent-1" type="UniRTSP"> <sdp-origin>UniMRCPServer</sdp-origin> <rtsp-ip>10.10.0.1</rtsp-ip> <rtsp-port>1554</rtsp-port> <resource-map> __MRCP_RESOURCE_MAP__ </resource-map> <max-connection-count>N</max-connection-count> <sdp-origin>UniMRCPServer</sdp-origin> </rtsp-uas></pre>

Atributos	Descrição
<rtsp-ip>	Especifica o endereço IP (agente RTSP), se não especificar,

	será utilizado o valor do parâmetro <ip>
<rtsp-ext-ip>	Especifica o IP sobre NAT (agente RTSP), se não especificar, assume o valor do parâmetro <ext-ip>
<rtsp-port>	Especifica a porta local do agente RTSP
<resource-map>	Especifica os recursos utilizados nas mensagens RTSP. (speechsynthesizer, speechrecognizer)
__MRCP_RESOURCE_MAP__	Campo preenchido automaticamente pelo <i>configure</i> . Não editar. Veja item 4.5.
<max-connection-count>	Especifica o número máximo de conexões TCP/RTSP por agente.
<sdp-origin>	Especifica o campo SDP de origem

Processamento de Mídia

Parâmetros	Descrição
<media-engine>	Especifica o mecanismo do processamento de mídia.
Código	<media-engine id="Media-Engine-1"> <realtime-rate>1</realtime-rate> </media-engine>

Atributos	Descrição
<realtime-rate>	Especifica a taxa/velocidade de trabalho do mecanismo, padrão = real-time = 1.

Protocolo de Mídia

Parâmetros	Descrição
<rtp-factory>	Especifica o comportamento do RTP, responsável pelo <i>streaming</i> do áudio.
Código	<rtp-factory id="RTP-Factory-1"> <rtp-ip>10.10.0.1</rtp-ip> <rtp-ext-ip>10.10.0.1</rtp-ext-ip> <rtp-port-min>5000</rtp-port-min> <rtp-port-max>6000</rtp-port-max> </rtp-factory>

Atributos	Descrição
<rtp-ip>	Especifica o endereço IP de mídia , se não especificar, será utilizado o valor do parâmetro <ip>
<rtp-ext-ip>	Especifica o IP sobre NAT de mídia , se não especificar, assume o valor do parâmetro <ext-ip>
<rtp-port-min>	Especifica o número da porta mínima permitido.
<rtp-port-max>	Especifica o número da porta máxima permitido.

Plugins do Servidor

Parâmetros	Descrição
<plugin-factory>	Especifica os mecanismos que serão carregados pelo servidor
Código	<pre><plugin-factory> __MRCP_PLUGIN_ENGINE__ </plugin-factory></pre>

Atributos	Descrição
__MRCP_PLUGIN_ENGINE__	Campo preenchido automaticamente pelo <i>configure</i> . Não editar. Veja item 4.5.

Codec de Áudio

Parâmetros	Descrição
<rtp-setting>	Especifica o comportamento dos codec de áudio.
Código	<pre><settings> <rtp-settings id="RTP-Settings-1"> <jitter-buffer> <adaptive>1</adaptive> <playout-delay>50</playout-delay> <max-playout-delay>600</max-playout-delay> <time-skew-detection>1</time-skew-detection> </jitter-buffer> <ptime>20</ptime> <codecs own-preference="false">PCMU PCMA</codecs> <rtpc enable="false"/> </rtp-settings> </settings></pre>

Atributos	Descrição
<jitter-buffer>	Especifica as configurações de Buffer relacionado ao <i>streaming</i> do áudio.
<ptime>	Especifica o tempo de empacotamento dos dados trafegados pela rede (mili segundos, ms).
<codecs>	Especifica os codecs de áudio suportados (PCMU, PCMA, L16/96/8000, telephone-event/101/8000)
<rtcp>	Especifica as configurações de controle (RTCP).

5.6 Parâmetros CPqD Texto Fala para MRCP

Os parametros abaixo podem ser alterados no arquivo:

- /opt/cpqd/mrcp/server/conf/cpqd/cpqd-tts.xml

Recursos

Parâmetros	Descrição
<resource id>	Especifica os mecanismos que serão carregados pelo servidor
Código	<resource id="speechsynth" enable="true" />

Atributos	Descrição
enable	Habilita/Desabilita recurso de sintese de fala. Valores: true/false

Mapa de recursos

Parâmetros	Descrição
<param name>	Especifica o nome do recurso para a versão MRCPv1.
Código	<param name="speechsynth" value="speechsynthesizer" />

Atributos	Descrição
value	Nome do recurso de media para MRCPv1

Engine de Texto Fala

Parâmetros	Descrição
<engine id>	Especifica o mecanismo e a biblioteca que serão utilizados pelo servidor.
Código	<pre><engine id="CPqD-TTS" name="cpqd_mrqp_tts" enable="true"> <param name="tts_conf" value="/opt/cpqd/tts/" /> <param name="tts_log" value="/opt/cpqd/tts/" /> <param name="tts_engine" value="/opt/cpqd/tts/libcpqdtts.so" /> </engine></pre>

Atributos	Descrição
<engine>	id="TTS" name="cpqd_mrqp_tts" enable="true"
name="tts_conf" value=	Especifica o diretório que contém os arquivos de configuração (.conf e .dat), as vozes (.voice) e a licença (.cpqd e .lic). Padrão: /opt/cpqd/tts/
name="tts_log" value=	Especifica o diretório que será gravado os logs para auditoria. Padrão: /opt/cpqd/tts/
name="tts_engine" value=	Especifica o diretório e o nome da biblioteca CPqD Texto Fala. Padrão: /opt/cpqd/tts/libcpqdtts.so

5.7 Parâmetros CPqD ASR para MRCP

Os parâmetros abaixo podem ser alterados no arquivo:

- /opt/cpqd/mrcp/server/conf/cpqd/cpqd-asr.xml

Recursos

Parâmetros	Descrição
<resource id>	Especifica os mecanismos que serão carregados pelo servidor
Código	<resource id="speechrecog" enable="true" />

Atributos	Descrição
enable	Habilita/Desabilita recurso de reconhecimento de fala. Valores: true/false

Mapa de recursos

Parâmetros	Descrição
<param name>	Especifica o nome do recurso para a versão MRCPv1.
Código	<param name="speechrecog" value="speechrecognizer"/>

Atributos	Descrição
value	Nome do recurso de media para MRCPv1

Engine de Reconhecimento de Fala

Parâmetros	Descrição
<engine id>	Especifica o mecanismo e a biblioteca que serão utilizados pelo servidor.
Código	<engine id="CPqD-ASR" name="cpqd_mrcp_asr" enable="true"> <param name="cpqd_asr_server" value="" /> <param name="timeout_net_sec" value="20" /> </engine>

Atributos	Descrição
<engine>	id="ASR" name="cpqd_mrcp_asr" enable="true"
name="cpqd_asr_server" value=	Especifica o endereçamento completo em que o servidor de reconhecimento ou o balanceador de carga está instalado. Padrão: ws://127.0.0.1:8025/asr-server/asr
name="timeout_net_sec" value=	Determina o tempo máximo de espera da conexão websocket entre o servidor ASR e servidor MRCP. Padrão: 20

5.8 Cenários de utilização

Veja os exemplos de configuração para cenários de utilização do Servidor CPqD MRCP.

MRCPv1

Cliente e Servidor MRCP localizado na mesma rede, veja a figura 5.1 .

- IP do servidor: 192.168.1.10
- porta RTSP do servidor: 1554
- faixa de portas RTP do servidor: 5000 a 6000
- codecs do servidor: PCMU, PCMA

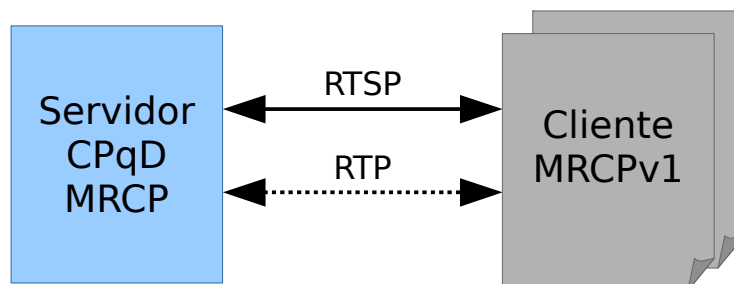


Figura 5.1: Cenário de utilização Cliente(s) MRCPv1

MRCPv2

Cliente e Servidor MRCP localizado na mesma rede, veja a figura 5.2 .

- IP do servidor: 192.168.1.10
- porta SIP do servidor: 8060
- porta MRCPv2 do servidor: 1544
- faixa de porta RTP do servidor: 5000 a 6000
- codecs do servidor: PCMU, PCMA

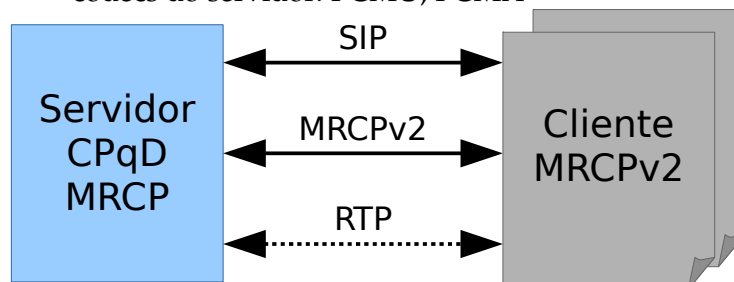
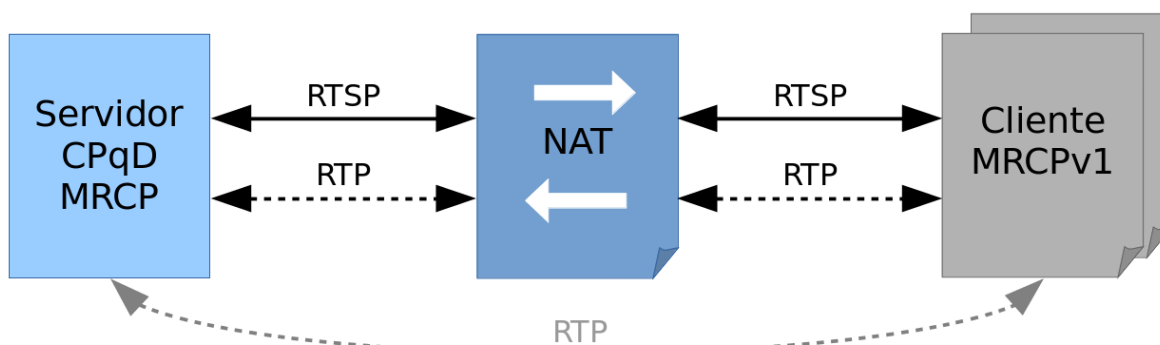


Figura 5.2: Cenário de utilização Cliente(s) MRCPv2

MRCPv1 com NAT

Servidor CPqD MRCP localizado atrás do NAT, veja a figura abaixo:

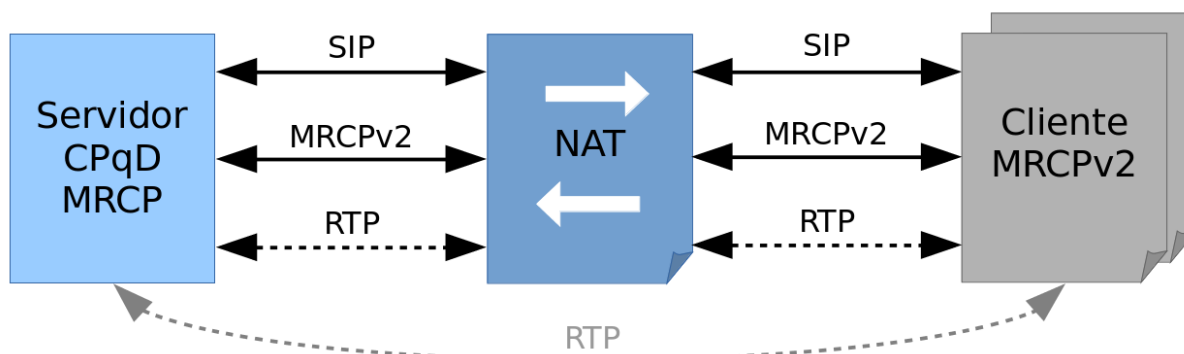
- IP do servidor: 192.168.1.10
- endereço IP interno do roteador: 192.168.1.1
- endereço IP externo do roteador: 10.10.1.1
- porta RTSP do servidor: 1554
- faixa de porta RTP do servidor: 5000 a 6000
- codecs do servidor: PCMU, PCMA



MRCPv2 com NAT

Servidor CPqD MRCP localizado atrás do NAT, veja a figura abaixo:

- IP do servidor: 192.168.1.10
- endereço IP interno do roteador: 192.168.1.1
- endereço IP externo do roteador: 10.10.1.1
- porta SIP do servidor: 8060
- porta MRCPv2 do servidor: 1544
- faixa de porta RTP do servidor: 5000 a 6000
- codecs do servidor: PCMU, PCMA



5.9 Logs do Servidor CPqD MRCP

Pela configuração padrão, apenas as informações relevantes (INFO) da comunicação MRCP serão escritas no arquivo armazenado em:

- /opt/cpqd/mrcp/server/log/unimrcpserver-xx.log

O arquivo abaixo descreve as configurações de LOG suportada pelo servidor CPqD MRCP. Qualquer alteração será válida somente após reiniciar o Servidor CPqD MRCP. A modificação pode acarretar em aumento significativo e crescente de espaço do disco (*hard drive*, HD).

- /opt/cpqd/mrcp/server/conf/logger.xml

6 PLATAFORMA DE VOZ (URA)

O Servidor CPqD MRCP é compatível com as plataformas de voz que suportam o padrão MRCP versão 1 e versão 2. Na maioria das situações, para integrar o Servidor MRCP com o cliente MRCP (plataforma de voz), basta configurar o endereço IP, porta SIP, porta MRCPv2, porta RTSP e o RTSP URL.

Em sua maioria, as plataformas de voz que suportam o padrão MRCP possuem um menu de configuração que permite ao administrador configurar onde a plataforma de voz deve procurar os recursos de reconhecimento de fala (**CPqD ASR**) e síntese de fala (**CPqD Texto Fala**).

É comum que as plataformas de voz apresentem configurações predefinidas para servidores de síntese e reconhecimento de fala específicos. Se o **CPqD ASR** e o **CPqD Texto Fala** não constam dentre as opções disponíveis, há geralmente uma opção de servidor genérico ou personalizado que pode ser selecionada e configurada; caso contrário, pode-se selecionar as configurações destinadas aos outros fornecedores, como IBM.

6.1 Servidor CPqD MRCP com o Avaya Voice Portal

Para configurar o Avaya Voice Portal 7.1, deve usar as ferramentas web da Avaya com a permissão de usuário e senha para realizar a configuração no portal de voz.

Adicionar reconhecimento de fala (**CPqD ASR**) ou síntese de fala (**CPqD Texto Fala**):

1. Após realizar o login, clicar em **Servidores de Voz**.
2. Selecionar a aba **ASR** ou **TTS** e clicar no botão Adicionar
3. Digitar um nome para o servidor
4. Pode-se deixar a maioria das configurações em seus valores padrão, mas o **Tipo de mecanismo** deve ser ajustado para **Nuance**
5. Digitar o **Endereço da rede** do **CPqD MRCPv1**
6. Digitar a **Porta base** do **CPqD MRCPv1** e Clicar em salvar
7. Necessário desenvolver um diálogo com o Avaya AAOD utilizando o Menu->Prompt->TTS e habilitar um plano de discagem para testes.

6.2 Servidor CPqD MRCP com o Freeswitch

Seguir o passo a passo para configurar o Freeswitch 1.2 (ou superior) com o Servidor CPqD MRCP.

1. Descomentar a linha do arquivo: <freeswitch-install>
/autoload_configs/modules.conf.xml

```
<load module="mod_unimrcp"/>
```

2. Especificar o profile padrão do CPqD MRCP no arquivo: <freeswitch-install>
/autoload_configs/unimrcp.conf.xml

```
<configuration name="unimrcp.conf" description="UniMRCP Client">
  <settings>
    <param name="default-tts-profile" value="cpqd-mrcp-v1"/>
    <param name="default-asr-profile" value="cpqd-mrcp-v2"/>
    <param name="log-level" value="INFO"/>
    <param name="enable-profile-events" value="false"/>
    <param name="max-connection-count" value="100"/>
    <param name="offer-new-connection" value="0"/>
    <param name="request-timeout" value="3000"/>
  </settings>
  <profiles>
    <X-PRE-PROCESS cmd="include" data="../mrcp_profiles/*.xml"/>
  </profiles>
</configuration>
```

3. Criar arquivo como: <freeswitch-install>/mrcp_profiles/cpqd-mrcp.xml
4. Adicionar os profiles MRCPv1 e MRCPv2 do CPqD MRCP conforme descrito abaixo:

```
<include>
  <!-- CPqD Speech Server -->
  <!-- MRCP V1 -->
  <profile name="cpqd-mrcp-v1" version="1">
    <param name="server-ip" value="Inserir IP do servidor MRCP"/>
    <param name="server-port" value="1554"/>
    <param name="resource-location" value="media"/>
    <param name="speechsynth" value="speechsynthesizer"/>
    <param name="speechrecog" value="speechrecognizer"/>
    <!--param name="rtp-ext-ip" value="auto"/-->
    <param name="rtp-ip" value="Inserir IP do cliente MRCP"/>
    <param name="rtp-port-min" value="4000"/>
    <param name="rtp-port-max" value="5000"/>
    <!--param name="playout-delay" value="50"/-->
    <!--param name="max-playout-delay" value="200"/-->
    <!--param name="ptime" value="20"/-->
    <param name="codecs" value="PCMU PCMA"/>
    <!-- Add any default MRCP params for SPEAK requests here -->
```

```

</profile>
<!-- MRCP V2 -->
<profile name="cpqd-mrcp-v2" version="2">
  <param name="client-ip" value="auto"/>
  <param name="client-port" value="5090"/>
  <param name="server-ip" value="Inserir IP do servidor MRCP"/>
  <param name="server-port" value="8060"/>
  <param name="sip-transport" value="tcp"/>
  <!--param name="rtp-ext-ip" value="auto"/-->
  <param name="rtp-ip" value="Inserir IP do cliente MRCP"/>
  <param name="rtp-port-min" value="4000"/>
  <param name="rtp-port-max" value="5000"/>
  <!--param name="playout-delay" value="50"/-->
  <!--param name="max-playout-delay" value="200"/-->
  <!--param name="ptime" value="20"/-->
  <param name="codecs" value="PCMU PCMA"/>
</profile>
</include>

```

5. Editar os campos descritos acima:

- “server-ip”: ip do Servidor CPqD MRCP
- “rtp-ip”: ip do Cliente MRCP (freeswitch) “rtp-ip”: ip do Cliente MRCP (freeswitch)

6. Exemplo de diálogo (linguagem Lua): <freeswitch-install>/dialplan/teste-tts.lua

```

session:answer()
session:sleep(300)
session:set_tts_params("unimrcp:cpqd-mrcp-v1", "rosana-highquality.voice")
status = session:speak("Bem Vindo ao CpqD!")
freeswitch.consoleLog("speak status: " .. status) -- 0=Erro genérico 1=Sucesso
session:hangup();

```

7. Exemplo de Reconhecimento e síntese de fala (linguagem Lua): <freeswitch-install>/dialplan/teste-asr-tts.lua

```

session:answer()
session:sleep(300)
session:set_tts_params("unimrcp:cpqd-mrcp-v2", "rosana-highquality.voice")
session:execute("play_and_detect_speech", "say:unimrcp:rosana-highquality.voice:Diga uma mensagem. detect:unimrcp {start-input-timers=false, define-grammar=true, no-input-timeout=10000} builtin:slm/general")
result = session:getVariable('detect_speech_result')
freeswitch.consoleLog("recognition result: " .. result)
session:hangup();

```

- Para executar o diálogo acima é necessário configurar um plano de discagem no Freeswitch para ouvir a síntese em um Softphone. (<freeswitch-install>/dialplan/)

6.3 Servidor CPqD MRCP com o Asterisk 13

Para integrar o Servidor CPqD MRCP com o Asterisk 13 utilizamos o protocolo MRCP. O cliente unimrcp implementa esse protocolo. Siga os passo abaixo para instalação e configuração:

Instalando o unimrcp-asterisk

Pre-requisitos:

- unimrcp deps 1.4
- unimrcp 1.4

Comando para instalar o unimrcp-asterisk:

- `$ cd /tmp/`
- `$ git clone https://github.com/unispeech/asterisk-unimrcp.git`
- `$ cd asterisk-unimrcp/`
- `$ git checkout asterisk-unimrcp-1.4.0`
- `$./bootstrap`
- `$./configure`

Se o configure não identificar a versão do asterisk:

- `$ asterisk -V`

Confirme a versão do asterisk acima, e adicione baixo:

- `$./configure --with-asterisk-version=13.14.0`
- `$ make`
- `$ sudo make install`

Para testar se o unimrcp client está instalado, reinicie o asterisk e na interface de comando (CLI) digite:

```
*CLI> module show like res_speech_unimrcp.so
Module                Description                Use Count
res_speech_unimrcp.so  UniMRCP Speech Engine      0
1 modules loaded
*CLI> module show like app_unimrcp.so
Module                Description                Use Count
app_unimrcp.so        MRCP suite of applications  0
1 modules loaded
```

Configuração do MRCP no Asterisk:

Como visto acima, o asterisk-unimrcp possui dois módulos:

- res_speech_unimrcp
 - Integração através da Generic Speech Recognition API do asterisk.
- app_unimrcp
 - Uma aplicação para ser usada no dialplan do asterisk.

Generic Speech Recognition API:

Adicionar as seguintes informações no arquivo: /etc/asterisk/res-speech-unimrcp.conf

```
unimrcp-profile = cpqd-mrcp-v2 ; CPqD MRCPv2 Server
unimrcp-profile = cpqd-mrcp-v1 ; CPqD MRCPv1 Server
```

Criar o arquivo em: /usr/local/unimrcp/conf/client-profiles/cpqd.xml com o seguinte conteúdo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- UniMRCP client document -->
<unimrcpclient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="1.0">
  <settings>
    <!-- SIP MRCPv2 settings -->
    <sip-settings id="UniMRCP-SIP-Settings">
      <server-ip>_SEVER_IP_</server-ip>
      <server-port>8060</server-port>
    </sip-settings>

    <!-- RTSP MRCPv1 settings -->
    <rtsp-settings id="UniMRCP-RTSP-Settings">
      <server-ip>_SEVER_IP_</server-ip>
      <server-port>1554</server-port>
      <resource-location>media</resource-location>
      <resource-map>
        <param name="speechrecog" value="speechrecognizer"/>
        <param name="speechsynth" value="speechsynthesizer"/>
      </resource-map>
    </rtsp-settings>
  </settings>

  <profiles>
    <!-- CPqD MRCPv2 profile -->
```



```
<mrcpv2-profile id="cpqd-mrcp-v2">
  <sip-uac>SIP-Agent-1</sip-uac>
  <mrcpv2-uac>MRCPv2-Agent-1</mrcpv2-uac>
  <media-engine>Media-Engine-1</media-engine>
  <rtp-factory>RTP-Factory-1</rtp-factory>
  <sip-settings>UniMRCP-SIP-Settings</sip-settings>
  <rtp-settings>RTP-Settings-1</rtp-settings>
</mrcpv2-profile>

<!-- CPqD MRCPv1 profile -->
<mrcpv1-profile id="cpqd-mrcp-v1">
  <rtsp-uac>RTSP-Agent-1</rtsp-uac>
  <media-engine>Media-Engine-1</media-engine>
  <rtp-factory>RTP-Factory-1</rtp-factory>
  <rtsp-settings>UniMRCP-RTSP-Settings</rtsp-settings>
  <rtp-settings>RTP-Settings-1</rtp-settings>
</mrcpv1-profile>
</profiles>
</unimrcpclient>
```

Unimrcp client Application:

Editar o arquivo: /etc/asterisk/mrcp.conf

Com profile default com segue abaixo:

```
[general]
; Default ASR and TTS profiles.
default-asr-profile = speech-cpqd-mrcp2
default-tts-profile = speech-cpqd-mrcp2
```

Adicione o profile CPqD MRCPv2 no arquivo:

```
;
; Profile for CPqD Speech Server MRCPv2
;
[speech-cpqd-mrcp2]
; MRCP settings
version = 2
;
; SIP settings
server-ip = <CPqD MRCP Server IP>
server-port = 8060
;
; SIP user agent
client-ip = <MRCP Client IP>
```

```
client-port = 8093
; SIP transport either UDP or TCP.
sip-transport = udp
;
; RTP factory
rtp-ip = <MRCP Client IP>
rtp-port-min = 4000
rtp-port-max = 5000
;
; Jitter buffer settings
playout-delay = 50
max-playout-delay = 200
;
; RTP settings
ptime = 20
codecs = PCMU PCMA L16/96/8000 telephone-event/101/8000
;
; RTCP settings
rtcp = 0
;rtcp = 1
;rtcp-bye = 2
;rtcp-tx-interval = 5000
;rtcp-rx-resolution = 1000
```

Adicione o profile CPqD MRCPv1 no arquivo:

```
;
; Profile for CPqD Speech Server MRCPv1
;
[speech-cpqd-mrcp1]
; MRCP settings
version = 1
;
; RTSP settings
server-ip = <CPqD MRCP Server IP>
server-port = 1554
resource-location = media
speechsynth = speechsynthesizer
speechrecog = speechrecognizer
;
; RTP factory
rtp-ip = <MRCP Client IP>
rtp-port-min = 4000
rtp-port-max = 5000
;
; Jitter buffer settings
```

```
playout-delay = 50
max-playout-delay = 200
;
; RTP settings
ptime = 20
codecs = PCMU PCMA L16/96/8000 telephone-event/101/8000
;
; RTCP settings
rtcp = 0
;rtcp = 1
;rtcp-bye = 2
;rtcp-tx-interval = 5000
;rtcp-rx-resolution = 1000
```

Exemplos de dialplan do Asterisk:

Este exemplo demonstra como usar a aplicação `SynthAndRecog()` com uma síntese de voz em texto plano e uma gramática built-in.

```
exten => s,1,Answer
exten => s,n,SynthAndRecog("Bem vindo ao CPqD!", "builtin:slm/general", t=5000)
exten => s,n,Verbose(1, ${RECOG_STATUS}, ${RECOG_COMPLETION_CAUSE}, ${RECOG_RESULT})
exten => s,n,Hangup
```

Este exemplo demonstra como usar a aplicação `MRCPrecog` com uma gramática built-in de pizza para fala.

```
[mrcprecog-app1]
exten => s,1,Answer
exten => s,n,MRCPrecog(builtin:grammar/samples/pizza, p=default)
exten => s,n,Verbose(1, ${RECOGSTATUS}, ${RECOG_COMPLETION_CAUSE}, ${RECOG_RESULT})
exten => s,n,Hangup
```

Este exemplo demonstra como usar a aplicação `MRCPSynth` com texto plano e MRCP profile padrão.

```
[mrcpsynth-app1]
exten => s,1,Answer
exten => s,n,MRCPSynth("Bem vindo ao CPqD!", p=default)
exten => s,n,Verbose(1, ${SYNTHSTATUS})
exten => s,n,Hangup
```