

# CPqD Texto Fala

## Manual de Programação

Versão do produto: 3.3

Edição do documento: 2.17 – Maio de 2010

---

---

# CPqD Texto Fala

## Manual de Programação

Versão do produto: 3.3

Edição do documento: 2.17 – Maio de 2010

Número de páginas: 28

---

© 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.

**Gerência de Serviços e Aplicações Multimídia – GSAM/DTS**

**Tel: (+55 19) 3705-6406 Fax: (+55 19) 3705-5868**

**CPqD – Centro de Pesquisa e Desenvolvimento em Telecomunicações**

Rod. Campinas–Mogi-Mirim, km 118,5 – SP340

13086-902 – Campinas – SP – Brasil

<http://www.cpqd.com.br>

---

## Histórico de edições

**MAIO DE 2004**

Edição 1.0, Manual de Utilização do Texto Fala

**MARÇO DE 2006**

Edição 2.0, Manual de Utilização do Texto Fala

**ABRIL DE 2006**

Edição 2.1, Manual de Utilização do Texto Fala

**MAIO DE 2006**

Edição 2.2, Manual de Utilização do Texto Fala

**AGOSTO DE 2006**

Edição 2.3, Manual de Utilização do Texto Fala

**AGOSTO DE 2006**

Edição 2.4, Manual de Utilização do Texto Fala

**OUTUBRO DE 2006**

Edição 2.5, Manual de Utilização do Texto Fala

**JANEIRO DE 2007**

Edição 2.6, Manual de Utilização do Texto Fala

**MAIO DE 2007**

Edição 2.7, Manual de Utilização do Texto Fala

**JULHO DE 2008**

Edição 2.8, CPqD Texto Fala – Manual de Programação

**SETEMBRO DE 2008**

Edição 2.9, CPqD Texto Fala – Manual de Programação

**OUTUBRO DE 2008**

Edição 2.10, CPqD Texto Fala – Manual de Programação

**DEZEMBRO DE 2008**

Edição 2.11, CPqD Texto Fala – Manual de Programação

**DEZEMBRO DE 2008**

Edição 2.12, CPqD Texto Fala – Manual de Programação

**MAIO DE 2009**

Edição 2.13, CPqD Texto Fala – Manual de Programação

**JULHO DE 2009**

Edição 2.14, CPqD Texto Fala – Manual de Programação

**JANEIRO DE 2010**

Edição 2.15, CPqD Texto Fala – Manual de Programação

**MAIO DE 2010**

Edição 2.16, CPqD Texto Fala – Manual de Programação

**MAIO DE 2010**

Edição 2.17, CPqD Texto Fala – Manual de Programação

---

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
1.1	Objetivo do documento.....	1
1.2	Texto Fala.....	1
1.3	Visão Geral do Sistema.....	1
<b>2</b>	<b>ARQUITETURA DO SISTEMA .....</b>	<b>2</b>
2.1	Módulo de pré-processamento.....	2
2.2	Módulo de conversão ortográfico-fonética.....	2
2.3	Módulo de síntese.....	2
2.4	Módulo de codificação.....	2
<b>3</b>	<b>INSTALAÇÃO .....</b>	<b>4</b>
<b>4</b>	<b>INSTRUÇÕES DE PROGRAMAÇÃO .....</b>	<b>5</b>
4.1	Ajustando o arquivo de configuração da biblioteca Texto Fala.....	7
4.2	Carregando e inicializando a biblioteca Texto Fala.....	9
4.3	Alocando um novo canal de conversão texto-fala.....	9
4.4	Obtendo e ajustando a configuração de síntese do Texto Fala.....	9
4.5	Usando um canal de conversão texto-fala para processar um texto completo.....	11
4.6	Usando um canal de conversão texto-fala para processar partes de um texto.....	13
4.7	Desalocando um canal de conversão texto-fala.....	16
4.8	Finalizando a biblioteca Texto Fala.....	17
4.9	Funções auxiliares.....	17
<b>5</b>	<b>APÊNDICE I – CÓDIGOS DE ERRO .....</b>	<b>19</b>
<b>LISTA DE FIGURAS</b>		
<b>FIGURA 2.1</b>	<b>ARQUITETURA DO SISTEMA .....</b>	<b>3</b>
<b>LISTA DE TABELAS</b>		
<b>TABELA 5.1</b>	<b>CÓDIGOS DE ERRO DO CPqD TEXTO FALA .....</b>	<b>19</b>

---

# 1 Introdução

---

## 1.1 Objetivo do documento

Este documento tem o objetivo de descrever a arquitetura do sistema, o procedimento de instalação e o funcionamento da biblioteca Texto Fala.

## 1.2 Texto Fala

**Texto Fala** é o nome do sistema de conversão texto-fala (text-to-speech, ou TTS, em inglês) desenvolvido pelo **CPqD**. Este software tem a capacidade de sintetizar sinais de fala a partir de texto escrito em português. Ele pode trabalhar com qualquer tipo de texto ASCII, desde sentenças simples até grandes conjuntos de dados.

O software foi projetado para aplicações de propósito geral e, devido à sua interface simples, pode ser facilmente integrado a qualquer tipo de aplicação. Ele oferece ainda a possibilidade de gerar fala sintetizada em tempo real e é capaz de trabalhar tanto em ambientes *single-threaded* como *multi-threaded*.

## 1.3 Visão Geral do Sistema

O sistema Texto Fala consiste de uma biblioteca de ligação dinâmica (*ctf.dll* em Windows ou *libcpqdtf.so* em Unix) que contém funções para inicialização, uso e finalização do mecanismo de conversão texto-fala. Esta biblioteca pode ser integrada a qualquer tipo de aplicação, independentemente da linguagem de programação utilizada.

As funções contidas na biblioteca geram fala sintética a partir de qualquer texto escrito em formato ASCII padrão. O sinal de fala gerado segue os padrões do Português falado no Brasil. Há versões da biblioteca capazes de gerar voz masculina ou feminina em cada um dos seguintes formatos: PCM linear (8 ou 16 bits/amostra), PCM *lei-A* (8 bits/amostra) e PCM *lei-μ* (8 bits/amostra), em 8, 11,025 ou 16 kHz, com ou sem cabeçalho WAV. O Texto Fala pode trabalhar com múltiplos canais simultâneos de conversão texto-fala operando em tempo real (i.e. o tempo necessário para sintetizar uma sentença será igual ou inferior ao tempo necessário para a reprodução do áudio correspondente). O número máximo de canais simultâneos depende da plataforma empregada.

Conforme a versão do Texto Fala, pode ser fornecida uma chave de hardware que deve ser conectada à porta paralela ou à porta USB do computador. Neste caso, o sistema só funcionará corretamente quando a chave de hardware estiver conectada à máquina.

---

## 2 Arquitetura do Sistema

---

O sistema é composto por quatro módulos, os quais desempenham tarefas essenciais para a conversão texto-fala. Esses módulos são:

### 2.1 Módulo de pré-processamento

O módulo de pré-processamento transforma o texto de entrada em um fluxo de palavras e símbolos de pontuação. Ele traduz números, símbolos especiais, abreviações, etc. em palavras.

### 2.2 Módulo de conversão ortográfico-fonética

Uma vez que o Texto Fala converte sentenças em fala, ele precisa determinar os alofones (unidades fonéticas básicas) que compõem cada sentença. Esta tarefa é realizada pelo módulo de conversão ortográfico-fonética, que consiste basicamente de regras de transformação de sequências de letras em sequências de unidades fonéticas.

### 2.3 Módulo de síntese

O módulo de síntese gera um sinal de fala a partir da lista de unidades fonéticas gerada pelos módulos anteriores. O sintetizador seleciona unidades acústicas apropriadas na sua base de dados e as processa, inserindo ritmo e entonação de modo a obter fala sintetizada de alta qualidade.

### 2.4 Módulo de codificação

A função desse módulo é converter a saída gerada para o formato apropriado (linear, *lei-A*, *lei-μ*):

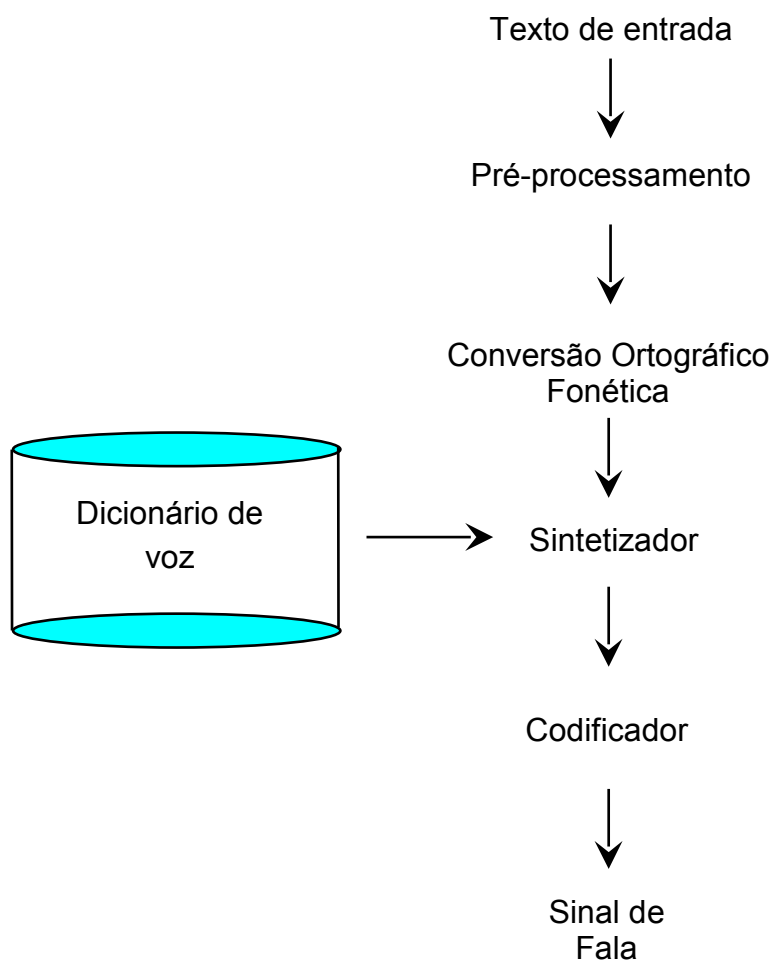


Figura 2.1 Arquitetura do Sistema.

---

## 3 Instalação

---

Este procedimento de instalação refere-se a um computador rodando na plataforma Windows.

Deve-se efetuar o *login* como ADMINISTRADOR, caso contrário o processo de instalação não funcionará adequadamente. Além disso, a chave de hardware, quando necessária, deve estar conectada à porta paralela ou USB do computador.

Para instalar o software do Texto Fala, basta executar o arquivo *setup.exe*, localizado no diretório \setup do CD-ROM de instalação. A instalação criará os arquivos necessários para o uso do sistema Texto Fala (arquivos executáveis e de dados).

O procedimento de instalação pode diferir do apresentado acima conforme a versão específica de distribuição do Texto Fala. Para maiores informações, consultar o arquivo "leiametext.txt" ou "leiametext.htm" contido no diretório raiz do CD-ROM de instalação.



---

## 4 Instruções de Programação

---

A biblioteca dinâmica Texto Fala (“ctf.dll” em Windows e “libcpqdtf.so” em Unix) contém as funções necessárias para executar o processo de conversão texto-fala. Estas funções encontram-se descritas a seguir:

- *tts\_inicializa()*: inicializa a biblioteca Texto Fala, devendo ser acionada antes de qualquer chamada às funções de síntese e de controle de canais.
- *tts\_finaliza()*: finaliza a biblioteca Texto Fala, devendo ser chamada quando a aplicação não mais necessitar das funções da biblioteca.
- *tts\_alocaCanal()*: aloca um novo canal de conversão texto-fala.
- *tts\_desalocaCanal()*: desaloca um canal de conversão texto-fala previamente alocado através de *tts\_alocaCanal()*, devendo ser chamada sempre que um dado canal não for mais utilizado pela aplicação.
- *tts\_obtemConfiguracao Sintese()*: obtém dados de configuração de síntese do Texto Fala.
- *tts\_ajustaConfiguracao Sintese()*: ajusta a configuração de síntese do Texto Fala.
- *tts\_sintetizaTexto\_ff()*: recebe um arquivo texto padrão em formato ASCII como entrada e gera como saída um arquivo contendo sinal de fala. Dependendo da versão da biblioteca utilizada, o arquivo gerado estará no formato PCM linear, *lei-A* ou *lei-μ* (com ou sem cabeçalho WAV).
- *tts\_sintetizaTexto\_mf()*: recebe uma cadeia de caracteres como entrada e gera como saída um arquivo contendo sinal de fala. Dependendo da versão da biblioteca utilizada, o arquivo gerado estará no formato PCM linear, *lei-A* ou *lei-μ* (com ou sem cabeçalho WAV).
- *tts\_sintetizaTexto\_fm()*: recebe um arquivo texto padrão em formato ASCII como entrada e produz amostras de fala como saída. Essas amostras serão escritas em uma área de memória alocada pela própria biblioteca Texto Fala. Dependendo da versão da biblioteca utilizada, o sinal de fala produzido estará no formato PCM linear, *lei-A* ou *lei-μ*. (com ou sem cabeçalho WAV). A área de memória alocada para o sinal pode ser liberada através da função *tts\_libera\_m()*.
- *tts\_sintetizaTexto\_mm()*: recebe uma cadeia de caracteres como entrada e produz amostras de fala como saída. Essas amostras serão escritas em uma área de memória alocada pela própria biblioteca Texto Fala. Dependendo da versão da biblioteca utilizada, o sinal de fala produzido estará no formato PCM linear, *lei-A* ou *lei-μ*. (com ou sem cabeçalho WAV). A área de memória alocada para o sinal pode ser liberada através da função *tts\_libera\_m()*.
- *tts\_sintetizaSentenca\_ff()*: recebe um arquivo texto padrão em formato

ASCII como entrada e gera como saída um arquivo contendo sinal de fala. Diferentemente da função *tts\_sintetizaTexto\_ff()*, *tts\_sintetizaSentenca\_ff()* processa uma única sentença do texto de entrada. Dependendo da versão da biblioteca utilizada, o arquivo gerado estará no formato PCM linear, *lei-A* ou *lei-μ* (com ou sem cabeçalho WAV).

- *tts\_sintetizaSentenca\_mf()*: recebe uma cadeia de caracteres como entrada e gera como saída um arquivo contendo sinal de fala. Diferentemente da função *tts\_sintetizaTexto\_mf()*, *tts\_sintetizaSentenca\_mf()* processa uma única sentença do texto de entrada. Dependendo da versão da biblioteca utilizada, o arquivo gerado estará no formato PCM linear, *lei-A* ou *lei-μ* (com ou sem cabeçalho WAV).
- *tts\_sintetizaSentenca\_fm()*: recebe um arquivo texto padrão em formato ASCII como entrada e produz amostras de fala como saída. Essas amostras serão escritas em uma área de memória alocada pela própria biblioteca Texto Fala. Diferentemente da função *tts\_sintetizaTexto\_fm()*, *tts\_sintetizaSentenca\_fm()* processa uma única sentença do texto de entrada. Dependendo da versão da biblioteca utilizada, o sinal de fala produzido estará no formato PCM linear, *lei-A* ou *lei-μ*. (com ou sem cabeçalho WAV). A área de memória alocada para o sinal pode ser liberada através da função *tts\_libera\_m()*.
- *tts\_sintetizaSentenca\_mm()*: recebe uma cadeia de caracteres como entrada e produz amostras de fala como saída. Essas amostras serão escritas em uma área de memória alocada pela própria biblioteca Texto Fala. Diferentemente da função *tts\_sintetizaTexto\_mm()*, *tts\_sintetizaSentenca\_mm()* processa uma única sentença do texto de entrada. Dependendo da versão da biblioteca utilizada, o sinal de fala produzido estará no formato PCM linear, *lei-A* ou *lei-μ*. (com ou sem cabeçalho WAV). A área de memória alocada para o sinal pode ser liberada através da função *tts\_libera\_m()*.
- *tts\_mensagemErro()*: retorna um ponteiro para uma cadeia de caracteres contendo a mensagem de erro correspondente ao último erro ocorrido em uma das funções da biblioteca Texto Fala.
- *tts\_versao()*: retorna um ponteiro para uma cadeia de caracteres contendo a versão da biblioteca Texto Fala.
- *tts\_libera\_m()*: libera uma área de memória alocada internamente à biblioteca Texto Fala.
- *tts\_libera\_f()*: apaga um arquivo gerado pela biblioteca Texto Fala.

No caso da versão do Texto Fala protegida por chave de hardware, o sistema só poderá ser iniciado com a chave conectada à porta paralela ou USB da máquina.

## 4.1 Ajustando o arquivo de configuração da biblioteca Texto Fala

A biblioteca Texto Fala possui um arquivo de configuração (“ctf.ini”) que pode ser modificado pelo usuário. Este arquivo pode se localizar no diretório onde se encontra instalada a biblioteca ou no diretório da aplicação que a utiliza; esta última opção é útil quando várias aplicações utilizam a biblioteca na mesma máquina com configurações diferentes. Caso o arquivo de configuração exista tanto no diretório da aplicação como no diretório de instalação do Texto Fala, o arquivo contido no diretório da aplicação tem prioridade.

A configuração é ajustada por meio da atribuição de valores a variáveis predefinidas. O formato deste arquivo é:

```
[TextoFala]
pathTTS=<diretório do texto fala>
pathDat=<diretório de dados>
pathReport=<diretório de logs>
codificacaoAudio=<codificação do áudio>
taxaAmostragem=<taxa de amostragem>
numeroCanaisAudio=<número de canais de áudio>
volume=<fator de volume>
ritmo=<fator de ritmo>
entonacao=<fator de entonação>
codificacaoTexto=<codificação do texto>
pausaInicial=<duração da pausa inicial>
pausaFinal=<duração da pausa final>
cabecalhoArquivo=<tipo de cabeçalho para síntese em arquivo>
cabecalhoMemoria=<tipo de cabeçalho para síntese em memória>
```

Linhas em branco são ignoradas no arquivo de configuração. Textos de comentário devem ser iniciados por dois caracteres de barra (“//”); todo o texto que segue esses caracteres é ignorado até o final da linha.

A ordem das variáveis no arquivo de configuração é irrelevante; no entanto, o indicador de seção “[TextoFala]” deve sempre ser inserido antes das variáveis. Os textos delimitados por “<” e “>” são apenas informativos e devem ser substituídos pelos valores das variáveis.

Os significados das variáveis e os respectivos valores permitidos encontram-se descritos a seguir:

- **pathTTS:** diretório da biblioteca Texto Fala.
- **pathDat:** diretório no qual se encontram os arquivos de dados (“.dat”) da biblioteca Texto Fala.
- **pathReport:** diretório no qual devem ser gerados os arquivos de log da biblioteca Texto Fala.
- **codificacaoAudio:** tipo de codificação do áudio sintetizado, podendo assumir os valores LINEAR\_16 (PCM linear, 16 bits/amostra), LINEAR\_8 (PCM linear, 8 bits/amostra), LEI\_A\_8 (PCM *lei-A*, 8 bits/amostra) e LEI\_MU\_8 (PCM *lei-μ*, 8 bits/amostra).

- **taxaAmostragem:** taxa de amostragem do áudio sintetizado, podendo assumir os valores KHZ\_8 (8 kHz), KHZ\_11 (11,025 kHz) e KHZ\_16 (16 kHz).
- **númeroCanaisAudio:** número de canais no áudio sintetizado, podendo assumir os valores MONO (sinal mono, isto é, 1 canal) e ESTEREO (sinal estéreo, isto é, 2 canais).
- **volume:** fator percentual de controle do volume do áudio sintetizado, variando de 0 (silêncio) a 200 (duas vezes o volume normal); um fator de 100 corresponde ao volume default. Este fator tem efeito linear sobre a amplitude do sinal, isto é, um aumento de duas vezes neste fator levará a um aumento de quatro vezes na potência do áudio.
- **ritmo:** fator percentual de controle do ritmo da fala sintetizada, variando de 30 (fala lenta) a 300 (fala rápida); um fator de 100 corresponde ao ritmo default.
- **entonacao:** fator percentual de controle da entonação da fala sintetizada, variando de 30 (fala grave) a 300 (fala aguda); um fator de 100 corresponde à entonação default.
- **codificacaoTexto:** tipo de codificação do texto de entrada, podendo assumir os valores ECS (ECS – *Extended Character Set*, usado no sistema operacional DOS), ISO (ISO 8859, usado no Windows e em algumas versões de Unix) e UTF\_8 (UTF-8 – *Unicode Transformation Format*, usado em algumas versões de Unix).
- **pausalInicial:** duração, em milissegundos, da pausa que é inserida no áudio antes da fala sintetizada, variando de 0 (início abrupto da fala) a 5000 (5 segundos de pausa antes da fala).
- **pausaFinal:** duração, em milissegundos, da pausa que é inserida no áudio após a fala sintetizada, variando de 0 (fim abrupto da fala) a 5000 (5 segundos de pausa após a fala).
- **cabecalhoArquivo:** tipo de cabeçalho que deve ser inserido no sinal de áudio quando este é gerado em arquivo, podendo assumir os valores SEM\_CAB (sem cabeçalho) e CAB\_WAVE (cabeçalho WAV padrão do Windows).
- **cabecalhoMemoria:** tipo de cabeçalho que deve ser inserido no sinal de áudio quando este é gerado em memória, podendo assumir os valores SEM\_CAB (sem cabeçalho) e CAB\_WAVE (cabeçalho WAV padrão do Windows).

Caso alguma variável seja omitida no arquivo de configuração, ela preserva seu valor default. Os valores default das variáveis de configuração são definidos na compilação da biblioteca Texto Fala, podendo variar conforme a aplicação-alvo. Em geral, a seguinte configuração default é empregada: pathTTS = "c:\ctf" para Windows ou pathTTS = "." para Unix, pathDat = "c:\ctf" para Windows ou pathDat = "." para Unix, pathReport = "c:\ctf\report" para Windows ou pathReport = "." para Unix, codificacaoAudio = LINEAR\_16, taxaAmostragem = KHZ\_16, numeroCanaisAudio = MONO, volume = 100, ritmo = 100, entonacao = 100, codificacaoTexto = ISO, pausalInicial = 30, pausaFinal = 50, cabecalhoArquivo = CAB\_WAVE, cabecalhoMemoria = CAB\_WAVE.

Em algumas versões específicas do Texto Fala, o recurso de arquivo de configuração pode estar desabilitado; neste caso, a biblioteca utiliza a configuração default definida na compilação. Ainda assim, é possível alterar a configuração via API do Texto Fala (ver seção *Obtendo e ajustando a configuração de síntese do Texto Fala* mais abaixo).

## 4.2 Carregando e inicializando a biblioteca Texto Fala

A aplicação principal deve carregar a biblioteca dinâmica Texto Fala (“ctf.dll” em Windows, “libcpqdtf.so” em Unix) e inicializá-la antes de poder acessar suas funções de síntese e de controle de canais. A inicialização da biblioteca é feita através da função *tts\_inicializa()*.

A sintaxe dessa função é:

***long tts\_inicializa ( )***

Essa função retorna um inteiro longo. Se o valor retornado for igual a zero, então a inicialização foi bem sucedida. Se o valor retornado for diferente de zero, então o processo de inicialização falhou. Nesse caso, o valor retornado corresponde a um código de erro que indica o problema ocorrido durante o processo de inicialização. A lista de possíveis erros está no Apêndice I, ao final deste documento.

## 4.3 Alocando um novo canal de conversão texto-fala

Antes de poder chamar as funções de síntese, um canal de conversão texto-fala deve ser alocado. Para tanto, deve-se usar a função *tts\_alocaCanal()*. O número máximo de canais que podem ser usados simultaneamente depende da versão específica do Texto Fala em uso.

A sintaxe dessa função é:

***long tts\_alocaCanal ( )***

Essa função retorna um inteiro longo. Se o valor retornado for maior ou igual a zero, então a alocação foi bem sucedida e o valor retornado corresponde ao *handle* do novo canal. Esse *handle* deve ser passado como parâmetro para as outras funções de conversão texto-fala. Se o valor retornado for menor que zero, então o processo de alocação falhou. Nesse caso, o valor retornado corresponde a um código de erro que indica o problema ocorrido durante o processo de alocação. A lista de possíveis erros está no Apêndice I, ao final deste documento.

## 4.4 Obtendo e ajustando a configuração de síntese do Texto Fala

Conforme explicado na seção *Ajustando o arquivo de configuração da biblioteca Texto Fala*, a configuração default de síntese do Texto Fala pode ser alterada através do arquivo de inicialização (“ctf.ini”).

A configuração de síntese pode também ser alterada de forma dinâmica por meio da função `tts_ajustaConfiguracaoSintese()`, podendo-se inclusive especificar configurações diferentes para cada canal de síntese. Todos os parâmetros definidos no arquivo de configuração podem ser alterados, com exceção dos diretórios `pathTTS`, `pathDat` e `pathReport`. A sintaxe dessa função é:

***long tts\_ajustaConfiguracaoSintese (long canal, long codificacaoAudio, long taxaAmostragem, long numeroCanaisAudio, long volume, long ritmo, long entonacao, long codificacaoTexto, long pausalnicial, long pausaFinal, long cabecalhoArquivo, long cabecalhoMemoria)***

onde *canal* é o canal de síntese ao qual se aplica a configuração. As descrições e valores possíveis dos demais parâmetros são idênticos aos apresentados na seção *Ajustando o arquivo de configuração da biblioteca Texto Fala*, exceto que os símbolos usados como parâmetros desta função (definidos no arquivo "bib\_textofala.h") levam o prefixo "TTS\_".

O parâmetro *canal* pode também corresponder a um dos seguintes valores: TTS\_TODO\_CANAIS, indicando que a alteração deve afetar todos os canais de síntese já alocados pertencentes ao processo, e TTS\_DEFAULT, indicando que a alteração se aplica à configuração de síntese default do Texto Fala. Ao alterar a configuração de síntese default, a configuração de canais já alocados não é afetada, mas novos canais alocados após a alteração seguirão a configuração indicada.

`tts_ajustaConfiguracaoSintese()` retorna um inteiro longo. Se o valor retornado for igual a zero, o ajuste da configuração foi bem sucedido. Se o valor retornado for diferente de zero, o ajuste de configuração falhou. Nesse caso, o valor retornado corresponde a um código de erro que indica o problema ocorrido durante o processo de ajuste de configuração. A lista de possíveis erros está no Apêndice I, ao final deste documento.

Caso a aplicação deseje conhecer a configuração de síntese do Texto Fala, deve ser usada a função `tts_obtemConfiguracaoSintese()`. Esta função tem a seguinte sintaxe:

***long tts\_obtemConfiguracaoSintese (long canal, long \*codificacaoAudio, long \*taxaAmostragem, long \*numeroCanaisAudio, long \*volume, long \*ritmo, long \*entonacao, long \*codificacaoTexto, long \*pausalnicial, long \*pausaFinal, long \*cabecalhoArquivo, long \*cabecalhoMemoria)***

onde *canal* é o canal de síntese cuja configuração se deseja conhecer. Os demais parâmetros são idênticos aos da função `tts_ajustaConfiguracaoSintese`, com exceção de que se trata de endereços de variáveis do tipo inteiro longo (de modo a permitir que os conteúdos dessas variáveis sejam alterados pela função).

O parâmetro *canal* pode também corresponder ao valor TTS\_DEFAULT, indicando que se deseja obter a configuração default do Texto Fala, e não a configuração de um canal específico.

`tts_obtemConfiguracaoSintese()` retorna um inteiro longo. Se o valor retornado for igual a zero, a configuração foi obtida com sucesso. Se o valor retornado for diferente de zero, a obtenção da configuração falhou. Nesse caso, o valor retornado corresponde a um código de erro que indica o problema ocorrido durante o processo de obtenção da configuração. A lista de possíveis erros está

no Apêndice I, ao final deste documento.

Todos os símbolos usados como valores de parâmetros na configuração do Texto Fala encontram-se definidos no arquivo “bib\_textofala.h”, fornecido juntamente com a biblioteca Texto Fala. Estas definições são: TTS\_LINEAR\_16 = 0L, TTS\_LINEAR\_8 = 1L, TTS\_LEI\_A\_8 = 2L, TTS\_LEI\_MU\_8 = 3L, TTS\_KHZ\_8 = 0L, TTS\_KHZ\_11 = 2L, TTS\_KHZ\_16 = 1L, TTS\_MONO = 0L, TTS\_ESTEREO = 1L, TTS\_SEM\_CAB = 0L, TTS\_CAB\_WAVE = 1L, TTS\_ECS = 0L, TTS\_ISO = 1L, TTS\_UTF\_8 = 2L, TTS\_DEFAULT = -9998L e TTS\_TODOS\_CANAIS = -9999L, onde a letra “L” após os números indica que se trata de inteiros longos.

Para facilitar o uso das funções de configuração de síntese, o arquivo “bib\_textofala.h” define ainda uma estrutura que agrupa todos os parâmetros de configuração de síntese do Texto Fala. Essa estrutura é associada ao tipo “\_tts\_config\_sintese” e contém os seguintes campos: codificacaoAudio, taxaAmostragem, numeroCanaisAudio, volume, ritmo, entonacao, codificacaoTexto, silencioInicial, silencioFinal, cabecalhoArquivo e cabecalhoMemoria. São definidas ainda duas macros que permitem obter e ajustar a configuração de síntese por meio dessa estrutura. São elas, respectivamente:

***long tts\_obtemConfigSintese (long canal, \_tts\_configSintese config)***

***long tts\_ajustaConfigSintese (long canal, \_tts\_configSintese config)***

onde *canal* é o canal de síntese cuja configuração se deseja obter ou ajustar, e *config* é a estrutura que receberá a configuração do canal ou que conterá a configuração a ser atribuída ao canal. No caso da macro *tts\_obtemConfigSintese()*, o parâmetro *canal* pode também assumir o valor TTS\_DEFAULT (obtenção da configuração de síntese default do Texto Fala). Já para a macro *tts\_ajustaConfigSintese*, o parâmetro *canal* pode também assumir os valores TTS\_DEFAULT (ajuste da configuração de síntese default do Texto Fala) e TTS\_TODOS\_CANAIS (ajuste simultâneo da configuração de síntese de todos os canais associados ao processo).

## 4.5 Usando um canal de conversão texto-fala para processar um texto completo

A biblioteca Texto Fala possui quatro funções que podem ser utilizadas para gerar sinal de fala a partir de um texto genérico: *tts\_sintetizaTexto\_ff()*, *tts\_sintetizaTexto\_mm()*, *tts\_sintetizaTexto\_mf()* e *tts\_sintetizaTexto\_fm()*. Uma vez criado um canal de conversão texto-fala através de *tts\_alocaCanal()*, pode-se utilizar esse canal quantas vezes forem necessárias a partir da chamada a qualquer uma dessas funções.

A sintaxe de *tts\_sintetizaTexto\_ff()* é:

***long tts\_sintetizaTexto\_ff (long canal, const char \*arquivoTexto, const char \*arquivoDeVoz)***

onde *canal* é o *handle* retornado por *tts\_alocaCanal()*, *arquivoTexto* é o nome do arquivo ASCII contendo o texto a ser sintetizado e *arquivoDeVoz* é o nome do arquivo de saída contendo o sinal de fala gerado.

O texto de entrada pode ser constituído de um ou mais parágrafos, cada um deles contendo uma ou mais sentenças. O número máximo de caracteres permitidos numa sentença é 10000, e em um parágrafo é 50000.

*tts\_sintetizaTexto\_ff()* retorna um inteiro longo. Um valor não-negativo indica que o processo de conversão texto-fala foi bem sucedido. Nesse caso, o valor retornado indica o tamanho, em bytes, do arquivo de fala gerado (incluindo um eventual cabeçalho WAV). Se o valor retornado for menor que zero, o processo falhou. Nesse caso, o valor retornado corresponde a um código de erro indicando o problema ocorrido. A lista de possíveis erros está no Apêndice I, ao final deste documento.

A segunda função de conversão texto-fala é *tts\_sintetizaTexto\_mm()*. Sua sintaxe é:

***long tts\_sintetizaTexto\_mm (long canal, const char \*buffTexto, char \*\*buffVoz)***

onde *canal* é o *handle* retornado por *tts\_alocaCanal()*, *buffTexto* é uma cadeia de caracteres (no formato ASCII) contendo o texto a ser sintetizado e *buffVoz* é o endereço de uma área de memória onde a função armazenará as amostras de fala sintetizada.

A cadeia de caracteres contendo o texto de entrada pode ser constituída de um ou mais parágrafos, cada um deles contendo uma ou mais sentenças. O número máximo de caracteres permitidos numa sentença é 10000, e em um parágrafo é 50000.

A área de memória onde serão armazenadas as amostras de fala sintetizada é alocada pela própria biblioteca, podendo ser posteriormente liberada pela aplicação através da função *tts\_libera\_m()*.

*tts\_sintetizaTexto\_mm()* retorna um inteiro longo. Um valor não-negativo indica que o processo de conversão texto-fala foi bem sucedido. Nesse caso, o valor retornado indica o tamanho, em bytes, do *buffer* de fala gerado (incluindo um eventual cabeçalho WAV). Se o valor retornado for menor que zero, então o processo de conversão texto-fala falhou. Nesse caso, o valor retornado corresponde a um código de erro indicando o problema ocorrido. A lista de possíveis erros está no Apêndice I, ao final deste documento.

A terceira função de conversão texto-fala é *tts\_sintetizaTexto\_mf()*. Sua sintaxe é:

***long tts\_sintetizaTexto\_mf (long canal, const char \*buffTexto, const char \*arquivoDeVoz)***

onde *canal* é o *handle* retornado por *tts\_alocaCanal()*, *buffTexto* é uma cadeia de caracteres (no formato ASCII) contendo o texto a ser sintetizado e *arquivoDeVoz* é o nome do arquivo de saída contendo o sinal de fala gerado.

A cadeia de caracteres contendo o texto de entrada pode ser constituída de um ou mais parágrafos, cada um deles contendo uma ou mais sentenças. O número máximo de caracteres permitidos numa sentença é 10000, e em um parágrafo é 50000.

*tts\_sintetizaTexto\_mf()* retorna um inteiro longo. Um valor não-negativo indica que o processo de conversão texto-fala foi bem sucedido. Nesse caso, o valor retornado indica o tamanho, em bytes, do arquivo de fala gerado (incluindo um



eventual cabeçalho WAV). Se o valor retornado for menor que zero, o processo falhou. Nesse caso, o valor retornado corresponde a um código de erro indicando o problema ocorrido. A lista de possíveis erros está no Apêndice I, ao final deste documento.

A quarta função de conversão é *tts\_sintetizaTexto\_fm()*. Sua sintaxe é:

***long tts\_sintetizaTexto\_fm (long canal, const char \*arquivoTexto, char \*\*buffVoz)***

onde *canal* é o *handle* retornado por *tts\_alocaCanal()*, *arquivoTexto* é o nome do arquivo ASCII contendo o texto a ser sintetizado e *buffVoz* é o endereço de uma área de memória onde a função armazenará as amostras de fala sintetizada.

O texto de entrada pode ser constituído de um ou mais parágrafos, cada um deles contendo uma ou mais sentenças. O número máximo de caracteres permitidos numa sentença é 10000, e em um parágrafo é 50000.

A área de memória onde serão armazenadas as amostras de fala sintetizada é alocada pela própria biblioteca, podendo ser posteriormente liberada pela aplicação através da função *tts\_libera\_m()*.

*tts\_sintetizaTexto\_fm()* retorna um inteiro longo. Um valor não-negativo indica que o processo de conversão texto-fala foi bem sucedido. Nesse caso, o valor retornado indica o tamanho, em bytes, do *buffer* de fala gerado (incluindo um eventual cabeçalho WAV). Se o valor retornado for menor que zero, então o processo de conversão texto-fala falhou. Nesse caso, o valor retornado corresponde a um código de erro indicando o problema ocorrido. A lista de possíveis erros está no Apêndice I, ao final deste documento.

## 4.6 Usando um canal de conversão texto-fala para processar partes de um texto

As funções acima descritas devem ser utilizadas preferencialmente em aplicações em que haja apenas um canal de conversão texto-fala, ou em aplicações com múltiplos canais em que os textos a serem processados não sejam muito extensos.

Caso o texto a ser processado seja muito extenso, a geração do sinal de fala pode levar vários segundos para ser finalizada. Para contornar esse efeito, a biblioteca Texto Fala possui outras quatro funções de conversão texto-fala: *tts\_sintetizaSentenca\_ff()*, *tts\_sintetizaSentenca\_mm()*, *tts\_sintetizaSentenca\_mf()* e *tts\_sintetizaSentenca\_fm()*. O funcionamento dessas funções é semelhante ao das funções anteriormente descritas. No entanto, ao invés de processar o texto inteiro, elas processam apenas uma sentença (frase). O tempo necessário para realizar a conversão de uma sentença é em geral bastante curto (da ordem de milissegundos).

Como essas funções processam apenas uma sentença por vez, é necessário chamá-las várias vezes sobre o mesmo texto de entrada, até que todo o texto tenha sido processado. Por isso, elas possuem um parâmetro a mais (*long \*offset*). O parâmetro *offset* indica a posição (em bytes) no arquivo ou cadeia de caracteres de entrada onde se inicia a sentença a ser processada. Ao retorno

da função, o valor de *offset* é automaticamente atualizado, passando a apontar para o início da próxima sentença. Caso não haja novas sentenças a processar, o valor atualizado de *offset* será dado pela constante EOT (definida como -99).

A sintaxe de *tts\_sintetizaSentenca\_ff()* é:

***long tts\_sintetizaSentenca\_ff (long canal, const char \*arquivoTexto, const char \*arquivoDeVoz, long \*offset)***

onde *canal* é o *handle* retornado por *tts\_alocaCanal()*, *arquivoTexto* é o nome do arquivo ASCII contendo o texto a ser sintetizado, *arquivoDeVoz* é o nome do arquivo de saída contendo o sinal de fala gerado e *offset* é um apontador para a posição (em bytes) no arquivo de entrada onde se inicia a sentença a ser sintetizada.

O texto de entrada pode ser constituído de um ou mais parágrafos, cada um deles contendo uma ou mais sentenças. O número máximo de caracteres permitidos numa sentença é 10000, e em um parágrafo é 50000.

*tts\_sintetizaSentenca\_ff()* retorna um inteiro longo. Um valor não-negativo indica que o processo de conversão texto-fala foi bem sucedido. Nesse caso, o valor retornado indica o tamanho, em bytes, do arquivo de fala gerado (incluindo um eventual cabeçalho WAV). Se o valor retornado for menor que zero, o processo falhou. Nesse caso, o valor retornado corresponde a um código de erro indicando o problema ocorrido. A lista de possíveis erros está no Apêndice I, ao final deste documento.

Caso a conversão tenha sido bem sucedida, o valor de *offset* é automaticamente atualizado, passando a apontar para o início da próxima sentença no texto de entrada. Caso não haja novas sentenças a processar, o valor atualizado de *offset* será EOT (-99).

É possível que o processo de conversão seja bem sucedido, mas que o sinal gerado seja vazio. Isso ocorre quando o texto de entrada estiver mal formado e contiver sentenças onde não haja nada a ser pronunciado. Nesses casos, a função retornará zero como tamanho do sinal gerado, indicando que, muito embora o processo de conversão tenha sido bem sucedido, o sinal gerado não é válido.

A segunda função capaz de processar uma sentença por vez é *tts\_sintetizaSentenca\_mm()*. Sua sintaxe é:

***long tts\_sintetizaSentenca\_mm (long canal, const char \*buffTexto, char \*\*buffVoz, long \*offset)***

onde *canal* é o *handle* retornado por *tts\_alocaCanal()*, *buffTexto* é uma cadeia de caracteres (no formato ASCII) contendo o texto a ser sintetizado, *buffVoz* é o endereço de uma área de memória onde a função armazenará as amostras de fala sintetizada e *offset* é um apontador para a posição (em bytes) na cadeia de caracteres de entrada onde se inicia a sentença a ser sintetizada.

A cadeia de caracteres contendo o texto de entrada pode ser constituída de um ou mais parágrafos, cada um deles contendo uma ou mais sentenças. O número máximo de caracteres permitidos numa sentença é 10000, e em um parágrafo é 50000.

A área de memória onde serão armazenadas as amostras de fala sintetizada é alocada pela própria biblioteca, podendo ser posteriormente liberada pela

aplicação através da função *tts\_libera\_m()*.

*tts\_sintetizaSentenca\_mm()* retorna um inteiro longo. Um valor não-negativo indica que o processo de conversão texto-fala foi bem sucedido. Nesse caso, o valor retornado indica o tamanho, em bytes, do *buffer* de fala gerado (incluindo um eventual cabeçalho WAV). Se o valor retornado for menor que zero, então o processo de conversão texto-fala falhou. Nesse caso, o valor retornado corresponde a um código de erro indicando o problema ocorrido. A lista de possíveis erros está no Apêndice I, ao final deste documento.

Caso a conversão tenha sido bem sucedida, o valor de *offset* é automaticamente atualizado, passando a apontar para o início da próxima sentença no texto de entrada. Caso não haja novas sentenças a processar, o valor atualizado de *offset* será EOT (–99).

É possível que o processo de conversão seja bem sucedido, mas que o sinal gerado seja vazio. Isso ocorre quando o texto de entrada estiver mal formado e contiver sentenças onde não haja nada a ser pronunciado. Nesses casos, a função retornará zero como tamanho do sinal gerado, indicando que, muito embora o processo de conversão tenha sido bem sucedido, o sinal gerado não é válido.

A terceira função capaz de processar uma sentença por vez é *tts\_sintetizaSentenca\_mf()*. Sua sintaxe é:

***long tts\_sintetizaSentenca\_mf (long canal, const char \*buffTexto, const char \*arquivoDeVoz, long \*offset)***

onde *canal* é o *handle* retornado por *tts\_alocaCanal()*, *buffTexto* é uma cadeia de caracteres (no formato ASCII) contendo o texto a ser sintetizado, *arquivoDeVoz* é o nome do arquivo de saída contendo o sinal de fala gerado e *offset* é um apontador para a posição (em bytes) na cadeia de caracteres de entrada onde se inicia a sentença a ser sintetizada.

A cadeia de caracteres contendo o texto de entrada pode ser constituída de um ou mais parágrafos, cada um deles contendo uma ou mais sentenças. O número máximo de caracteres permitidos numa sentença é 10000, e em um parágrafo é 50000.

*tts\_sintetizaSentenca\_mf()* retorna um inteiro longo. Um valor não-negativo indica que o processo de conversão texto-fala foi bem sucedido. Nesse caso, o valor retornado indica o tamanho, em bytes, do arquivo de fala gerado (incluindo um eventual cabeçalho WAV). Se o valor retornado for menor que zero, o processo falhou. Nesse caso, o valor retornado corresponde a um código de erro indicando o problema ocorrido. A lista de possíveis erros está no Apêndice I, ao final deste documento.

Caso a conversão tenha sido bem sucedida, o valor de *offset* é automaticamente atualizado, passando a apontar para o início da próxima sentença no texto de entrada. Caso não haja novas sentenças a processar, o valor atualizado de *offset* será EOT (–99).

É possível que o processo de conversão seja bem sucedido, mas que o sinal gerado seja vazio. Isso ocorre quando o texto de entrada estiver mal formado e contiver sentenças onde não haja nada a ser pronunciado. Nesses casos, a função retornará zero como tamanho do sinal gerado, indicando que, muito embora o processo de conversão tenha sido bem sucedido, o sinal gerado não

é válido.

A quarta função capaz de processar uma sentença por vez é *tts\_sintetizaSentenca\_fm()*. Sua sintaxe é:

***long tts\_sintetizaSentenca\_fm (long canal, const char \*arquivoTexto, char \*\*buffVoz, long \*offset)***

onde *canal* é o *handle* retornado por *tts\_alocaCanal()*, *arquivoTexto* é o nome do arquivo ASCII contendo o texto a ser sintetizado, *buffVoz* é o endereço de uma área de memória onde a função armazenará as amostras de fala sintetizada e *offset* é um apontador para a posição (em bytes) no arquivo de entrada onde se inicia a sentença a ser sintetizada.

O texto de entrada pode ser constituído de um ou mais parágrafos, cada um deles contendo uma ou mais sentenças. O número máximo de caracteres permitidos numa sentença é 10000, e em um parágrafo é 50000.

A área de memória onde serão armazenadas as amostras de fala sintetizada é alocada pela própria biblioteca, podendo ser posteriormente liberada pela aplicação através da função *tts\_libera\_m()*.

*tts\_sintetizaSentenca\_fm()* retorna um inteiro longo. Um valor não-negativo indica que o processo de conversão texto-fala foi bem sucedido. Nesse caso, o valor retornado indica o tamanho, em bytes, do *buffer* de fala gerado (incluindo um eventual cabeçalho WAV). Se o valor retornado for menor que zero, então o processo de conversão texto-fala falhou. Nesse caso, o valor retornado corresponde a um código de erro indicando o problema ocorrido. A lista de possíveis erros está no Apêndice I, ao final deste documento.

Caso a conversão tenha sido bem sucedida, o valor de *offset* é automaticamente atualizado, passando a apontar para o início da próxima sentença no texto de entrada. Caso não haja novas sentenças a processar, o valor atualizado de *offset* será EOT (-99).

É possível que o processo de conversão seja bem sucedido, mas que o sinal gerado seja vazio. Isso ocorre quando o texto de entrada estiver mal formado e contiver sentenças onde não haja nada a ser pronunciado. Nesses casos, a função retornará zero como tamanho do sinal gerado, indicando que, muito embora o processo de conversão tenha sido bem sucedido, o sinal gerado não é válido.

## 4.7 Desalocando um canal de conversão texto-fala

Quando a aplicação não tiver mais necessidade de usar um dado canal de conversão texto-fala, este deverá ser desalocado. Isso é feito por meio da função *tts\_desalocaCanal()*, a qual desaloca todas as estruturas de memória utilizadas pelo canal. A sintaxe dessa função é:

***long tts\_desalocaCanal (long canal)***

onde *canal* é o *handle* (gerado por *tts\_alocaCanal()*) do canal a ser desalocado. O parâmetro *canal* pode também assumir o valor TTS\_TODOS\_CANAIS (definido como -9999L, onde "L" sinaliza um inteiro longo), indicando que todos os canais pertencentes ao processo devem ser desalocados.

Essa função retorna um inteiro longo. Se o valor retornado for igual a zero, então a desalocação foi bem sucedida. Se o valor retornado for diferente de zero, então o processo de desalocação falhou. Nesse caso, o valor retornado corresponde a um código de erro que indica o problema ocorrido durante o processo de desalocação. A lista de possíveis erros está no Apêndice I, ao final deste documento.

## 4.8 Finalizando a biblioteca Texto Fala

Quando a aplicação não tiver mais necessidade das funções da biblioteca Texto Fala, esta deve ser finalizada. Isso é feito por meio da função *tts\_finaliza()*, a qual desaloca todas as estruturas de memória utilizadas pela biblioteca. A sintaxe dessa função é:

***long tts\_finaliza ()***

Essa função retorna um inteiro longo. Se o valor retornado for igual a zero, então a finalização foi bem sucedida. Se o valor retornado for diferente de zero, então o processo de finalização falhou. Nesse caso, o valor retornado corresponde a um código de erro que indica o problema ocorrido durante o processo de finalização. A lista de possíveis erros está no Apêndice I, ao final deste documento.

Caso haja canais de conversão texto-fala alocados no momento da chamada a *tts\_finaliza()*, estes são automaticamente desalocados.

## 4.9 Funções auxiliares

A função *tts\_mensagemErro()* pode ser utilizada para depurar os códigos de erro retornados pelas funções do Texto Fala acima descritas. Sua sintaxe é:

***char \*tts\_mensagemErro ()***

Esta função retorna um ponteiro para uma cadeia de caracteres contendo a mensagem de erro correspondente ao último erro ocorrido em uma das funções da biblioteca Texto Fala. A área de memória onde se encontra essa cadeia de caracteres é alocada internamente à biblioteca e, embora possa ser lida pela aplicação, não deve ser alterada nem desalocada por ela. Essa função serve apenas como auxílio na identificação de erros, mas não possui qualquer mecanismo de tratamento de erros. Cabe à aplicação realizar o tratamento apropriado para cada possível erro.

A função *tts\_versao()* retorna um ponteiro para uma cadeia de caracteres contendo a versão da biblioteca Texto Fala. Sua sintaxe é:

***char \*tts\_versao ()***

A área de memória onde se encontra essa cadeia de caracteres é alocada internamente à biblioteca e, embora possa ser lida pela aplicação, não deve ser alterada nem desalocada por ela.

Ex.: “CPqD Texto Fala 3.20 32x 16c 16000Hz” (Texto Fala versão 3.20; desempenho de 32 segundos de fala para cada segundo de processamento; 16

canais de síntese simultâneos; base de voz a 16 kHz).

A função *tts\_libera\_m()* libera uma área de memória alocada pela biblioteca Texto Fala. Sua sintaxe é:

***long tts\_libera\_m (void \*blocoMemoria)***

onde *blocoMemoria* é um ponteiro para o início do bloco a ser desalocado. Blocos de memória alocados pela biblioteca (por exemplo, *buffers* de fala) devem ser desalocados através de *tts\_libera\_m()*, e nunca por meio das instruções de desalocação de memória providas pela linguagem na qual for elaborada a aplicação. Em caso de execução bem sucedida, a função retorna zero; em caso de erro, a função retorna um código de erro não nulo.

A função *tts\_libera\_f()* apaga um arquivo gerado pela biblioteca Texto Fala. Sua sintaxe é:

***long tts\_libera\_f (const char \*nomeArquivo)***

onde *nomeArquivo* é uma string contendo o caminho para o arquivo a apagar. Em caso de execução bem sucedida, a função retorna zero; em caso de erro, a função retorna um código de erro não nulo.

## 5 Apêndice I – Códigos de erro

Tabela 5.1 Códigos de erro do CPqD Texto Fala.

Código	Descrição
-7	Arquivo com o texto de entrada está vazio ou não existe
-8	Não foi possível abrir o arquivo de depuração
-9	Tentativa de finalizar conversor texto-fala inexistente
-10	Erro na alocação de canal (não existem canais disponíveis)
-11	A senha passada para a função de inicialização é inválida
-12	Não foi possível alocar memória para carregar os recursos do conversor texto-fala
-13	Não foi possível alocar memória para o texto de entrada
-14	Não foi possível abrir o arquivo de saída de voz
-15	Não foi possível escrever no arquivo de saída de voz
-16	Não foi possível abrir o arquivo texto de entrada
-17	O parágrafo a ser processado é maior que o tamanho máximo permitido (8192 caracteres)
-18	Tentativa de uso do conversor texto-fala sem que este tenha sido inicializado
-19	Não foi possível abrir o arquivo “regtempo.log”
-20	Erro ao iniciar o filtro FIR
-21	Erro na alocação de memória (conversão 16 kHz para 8 kHz)
-22	Erro na alocação de memória (conversão 16 kHz para 8 kHz)
-23	Erro na alocação de memória (conversão 16 kHz para 8 kHz)
-24	Erro na alocação de memória (conversão lei-A)
-25	Erro na alocação de memória (conversão lei-A)
-26	Erro na alocação de memória (conversão lei-A)
-27	Erro na alocação de memória (conversão lei-A)
-28	Não foi possível abrir o arquivo de backup
-29	Não foi possível escrever no arquivo de backup
-30	Não foi possível abrir o arquivo 15.dat
-31	Não foi possível ler do arquivo 15.dat
-32	Não foi possível escrever no arquivo 15.dat
-33	Não foi possível abrir arquivo genérico em disco

-34	Não foi possível abrir arquivo genérico em RAM
-35	Erro de alocação de memória para arquivo em RAM
-36	Erro de alocação de memória para arquivo em RAM
-37	Não é possível abrir arquivo em disco para cópia de dados da memória
-38	Não é possível copiar dados da memória para arquivo em disco
-39	Espaço insuficiente em memória para escrita de dados
-40	Erro de alocação de memória para buffer genérico
-41	Erro na geração da sequência de polifones
-42	Tentativa de uso de um canal que já está sendo utilizado
-43	Tentativa de uso de um canal cujo descritor não está na faixa de canais válidos
-44	Tentativa de uso de um canal não iniciado
-45	Tentativa de desalocar um canal não iniciado
-46	Erro na alocação de memória para síntese de fala
-47	Não foi possível carregar a biblioteca que contém o software de proteção da chave de hardware (C50032.DLL)
-48	Não foi possível carregar a rotina de proteção da biblioteca da chave de hardware (C50032.DLL)
-49	Falha no C500
-50	Falha no C500
-51	Falha no C500
-52	Falha no C500
-53	Falha no C500
-54	Falha no C500
-55	Falha no C500
-56	Não foi possível copiar dados para buffer em memória
-57	Não foi possível inicializar de dados (9.dat)
-58	Não foi possível inicializar de dados (2.dat)
-59	Não foi possível abrir o arquivo 1.dat
-60	Não foi possível abrir o arquivo 2.dat
-61	Não foi possível abrir o arquivo 8.dat
-62	Erro de alocação de memória para o módulo de pré-processamento
-63	Não foi possível abrir o arquivo 11.dat ou 16.dat
-64	Erro de alocação de memória no módulo de conversão ortográfico-fonética
-65	Não foi possível abrir o arquivo 3.dat



-66	Não foi possível abrir o arquivo 12.dat
-67	Não foi possível abrir o arquivo 6.dat
-68	Não foi possível abrir o arquivo 7.dat
-69	Não foi possível abrir o arquivo 10.dat
-70	A memória alocada para síntese de fala é insuficiente
-71	Atualize a data do seu computador
-72	Utilização não autorizada. Contate o CPqD.
-73	O prazo de utilização da versão DEMO expirou. Contate o CPqD.
-74	Versão incorreta do programa. Contate o CPqD.
-75	Não foi possível abrir o arquivo 4.dat
-76	Não conseguiu alocar memória para frase a ser sintetizada
-77	Não conseguiu se posicionar no texto de entrada para extrair próxima sentença
-78	A sentença extraída do texto de entrada é uma sentença vazia
-79	O tipo de entrada especificado no arquivo descritor de configurações não é válido
-80	O tipo de prosódia especificado no arquivo descritor de configurações não é válido
-81	O tipo de língua especificado no arquivo descritor de configurações não é válido
-82	O gênero especificado no arquivo descritor de configurações não é válido
-83	A taxa de amostragem especificado no arquivo descritor de configurações não é válida
-84	O tipo de síntese especificado no arquivo descritor de configurações não é válido
-85	O tamanho da frase a ser sintetizada é maior que o tamanho máximo permitido
-86	A chave de hardware não está habilitada para funcionar com esta versão do Texto Fala
-87	Erro em alocação de memória
-88	Tentativa de exceder o comprimento máximo permitido para a transcrição fonética
-89	Erro de leitura no arquivo de sinapses
-90	O número máximo de chamadas de função da versão DEMO foi ultrapassado
-91	Erro na abertura do arquivo 19.dat
-92	O limite de tempo de utilização da versão DEMO foi ultrapassado
-93	O fator de modificação de duração está fora da faixa permitida
-94	O fator de modificação de F0 está fora da faixa permitida

-95	Não conseguiu criar região de memória compartilhada
-96	Falha no C500
-99	Chegou ao final do texto de entrada
-100	Erro indeterminado
-300	Erro em abertura de arquivo
-301	Erro em fechamento de arquivo
-302	Erro de leitura em arquivo
-303	Erro de escrita em arquivo
-304	Erro em alocação de memória
-305	Erro no arquivo de dicionário de unidades
-306	Símbolo fonético inválido
-307	Erro no arquivo de dicionário de unidades
-308	Número inválido
-309	Transcrição fonética truncada
-310	Erro de leitura no léxico do conversor ortográfico-fonético
-311	Erro no arquivo de dicionário de unidades
-312	Erro no arquivo de dicionário de unidades
-313	Erro no arquivo de seleção de unidades
-314	Fim de arquivo inesperado
-315	Erro no arquivo de seleção de unidades
-316	Erro no arquivo de dicionário de unidades
-317	Erro no arquivo de configuração
-318	Texto Fala não inicializado
-319	Limite de canais atingido
-320	Canal inválido
-321	Canal não alocado
-322	Canal em uso
-323	Número máximo de instâncias excedido
-324	Número máximo de classes fonéticas excedido
-325	Erro na criação da área de memória comum do sintetizador
-326	Erro no ajuste prosódico
-327	Erro em criação de mutex
-328	Erro em liberação de mutex
-329	Erro em obtenção de mutex
-330	Número máximo de processos excedido
-331	Classes fonéticas não inicializadas
-332	Caminho de diretório inválido
-333	Número máximo de unidades pré-selecionadas excedido
-334	Lista de marcas de segmentação vazia
-335	Lista de marcas de pitch vazia
-336	Tentativa de liberação de instância inexistente do Texto Fala
-337	Offset de quadro inválido

-338	Erro em manipulação de arquivo
-339	Energia de quadro inválida
-340	Erro de acesso ao registro
-341	Erro de timeout em mutex
-342	Erro em criação de região crítica
-343	Erro em finalização de região crítica
-344	Erro em obtenção de região crítica
-345	Erro em liberação de região crítica
-346	Erro em finalização de mutex
-347	Handle de contador de tempo inválido
-348	Funções de medição de tempo não inicializadas
-349	String de mensagem de erro não alocada
-350	Erro na criação da área de memória comum do dicionário
-351	Não foi possível carregar o software de proteção da chave de hardware SENTINEL
-352	Não foi possível carregar o software de proteção da chave de hardware SENTINEL
-353	Não foi possível carregar o software de proteção da chave de hardware SENTINEL
-354	Falha no SENTINEL
-355	Falha no SENTINEL
-356	Falha no SENTINEL
-357	Falha no SENTINEL
-358	Falha no SENTINEL
-359	Falha no SENTINEL
-360	Falha no SENTINEL
-361	Falha no SENTINEL
-362	Falha na inicialização do Texto Fala
-363	Falha na criação do arquivo de requisição
-364	Arquivo de licença não encontrado
-365	Arquivo de licença inválido
-366	Código de licença inválido
-367	Código de licença bloqueado
-368	Arquivo de pré-seleção (25.dat) inválido
-369	Formato de codificação de áudio inválido
-370	Taxa de amostragem de saída inválida
-371	Tipo de cabeçalho de áudio inválido para arquivo
-372	Tipo de cabeçalho de áudio inválido para memória
-373	Identificador de conjunto de caracteres inválido
-374	Fator de volume fora da faixa válida
-375	Duração inválida para a pausa inicial da síntese
-376	Duração inválida para a pausa final da síntese
-377	Número de canais de áudio inválido
-378	Função não disponível nesta versão do Texto Fala

-379	Erro em criação de diretório
-380	Erro ao remover arquivo
-381	Exceção do sistema operacional
-382	Falha de conexão socket
-383	Falha de handshake ou handshake inválido
-384	Erro ao enviar semente para servidor
-385	Erro ao receber código para habilitar licença
-386	Código de licença inválido
-387	Arquivo de chave pública não encontrado
-388	Assinatura de timestamp inválida
-399	Resposta enviada sem desafio gerado
-401	Falha no uso da chave pública em verificação de assinatura
-402	Resposta ao desafio incorreta
-403	Tentativa de inicialização sem validação de licença
-404	Falha em conversão de áudio para o formato VOX
-405	Falha nas regras de pré-processamento
-406	Codificação do dicionário inválida
-407	Silêncio inválido no dicionário
-408	Erro interno do Texto Fala
-410	Resposta enviada a JCTF sem que o desafio tenha sido gerado
-411	Resposta incorreta ao desafio para ativação da JCTF
-412	Erro ao carregar a biblioteca Texto Fala
-413	Falha no gerenciamento de threads