

## Projeto 2026

### *Identificação de dígitos através de características extraídas de sinais de áudio*

**Enquadramento:** *Na atualidade, o reconhecimento e interpretação de voz humana está disponível em praticamente todos os dispositivos. Por exemplo, é comum os telemóveis/computadores realizarem tarefas ativadas por comandos de voz. A fase inicial de qualquer tarefa de reconhecimento passa pela aquisição e extração de características do sinal áudio. O objetivo principal é que estas características permitam discriminar diferentes sons, correspondentes a diferentes comandos de voz.*

**Objetivo:** *Este projeto visa a análise nos domínios da frequência e do tempo de sinais áudio com o objetivo de identificar os dígitos em inglês entre 0 e 9.*

**Introdução**<sup>1</sup>: A Figura 1 mostra uma ilustração do sistema de produção da fala humana. Os principais componentes anatómicos deste sistema são os pulmões, traqueia, laringe (órgão de produção da fala), cavidade faríngea (garganta), cavidade bucal e cavidade nasal. Existem também outros componentes anatómicos que contribuem para a produção da fala, como as pregas vocais (ou cordas), língua, lábios, dentes e mandíbula. Estes últimos são referidos como articuladores e movem-se para diferentes posições a fim de produzir vários sons da fala como vogais, consoantes ou outras unidades fonológicas de uma língua.

O processo de produção da fala humana pode ser resumido da seguinte forma. Durante a fala, o ar expelido dos pulmões segue para a traqueia e sobe para a glote, onde o fluxo é periodicamente interrompido pelo movimento das cordas vocais. A tensão das cordas vocais é ajustada pela laringe causando as vibrações das cordas. Dentro do trato vocal, também são produzidas constrições à passagem do ar através da língua, lábios e nariz. Estas mudanças nas configurações do sistema de fala humana, irão fazer com que as vibrações emitidas através dos lábios para o ar circundante (i.e., o som) possuam componentes de frequência diferentes.

A produção da fala pode ser vista como uma operação de filtragem, na qual as três cavidades principais do sistema de produção da fala constituem o filtro acústico principal.

<sup>1</sup> [https://link.springer.com/referenceworkentry/10.1007/978-0-387-73003-5\\_199](https://link.springer.com/referenceworkentry/10.1007/978-0-387-73003-5_199)

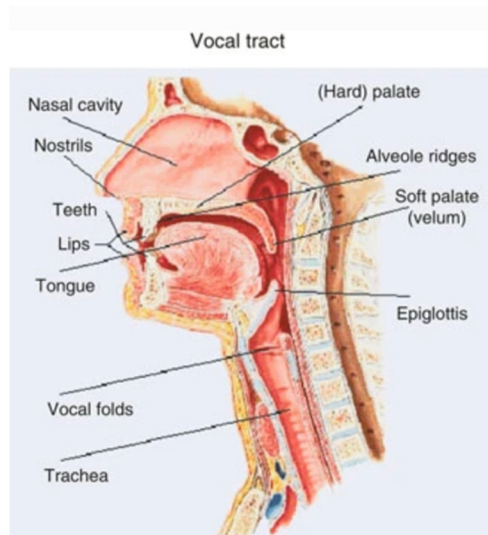


Fig 1: Componentes principais do sistema de produção de fala humana. Fonte: [https://link.springer.com/referenceworkentry/10.1007/978-0-387-73003-5\\_199](https://link.springer.com/referenceworkentry/10.1007/978-0-387-73003-5_199).

Linguagem de Programação: **MATLAB** (alternativamente Python).

Organização: **Grupos de dois alunos**, sempre que possível, e preferencialmente da mesma turma PL ou docente.

Dados: **Audio MNIST**, disponível em <https://www.kaggle.com/datasets/sripaadsrinivasan/audio-mnist/download?datasetVersionNumber=1>. Deve descompactar o ficheiro e considerar somente os sinais em bruto disponíveis na pasta *data*.

Descrição dos dados: Os dados fornecidos correspondem a sinais de voz emitidos por 60 participantes (cada uma das subdiretorias na pasta *data* contém os sinais correspondentes a 1 participante). Cada participante repetiu 50 vezes cada um dos dígitos, ou seja, cada uma das 60 pastas contém 500 sinais de áudio em formato *.wav*. Cada sinal de áudio foi adquirido a uma taxa de amostragem de 48000 Hz em modo mono-canal. Mais detalhes sobre os dados podem ser obtidos nas ligações seguintes.

<https://github.com/soerenab/AudioMNIST>

<https://www.kaggle.com/datasets/sripaadsrinivasan/audio-mnist>

<https://arxiv.org/abs/1807.03418>.

Estrutura: O trabalho está dividido em duas metas, com as seguintes datas de entrega:

**Meta 1:** 09/abr/2026, 23:59

**Meta 2:** 14/mai/2026, 23:59

Até à data estipulada para conclusão de cada meta, o grupo deverá submeter o trabalho correspondente no Inforestudante. Se não for possível cumprir com a data da Meta 1, o trabalho poderá ser incluído na submissão da Meta 2, no entanto com uma penalização automática de 50%.

Materiais a entregar: É altamente recomendável entregar de forma integrada código, figuras e texto, organizados por secções de acordo com as questões/exercícios do enunciado. Para isto é sugerida a utilização dos *Live Scripts (MATLAB)* ou, alternativamente, dos Jupyter Notebooks (Python/MATLAB). Deverá também entregar em cada meta o respetivo **ficheiro ‘.mat’** (ficheiro binário MATLAB que permite guardar o *workspace*, permitindo uma rápida recuperação e carregamento de todas as variáveis). Ao nível do código, sugere-se igualmente o uso de caminhos relativos para ficheiros/pastas, ao invés de absolutos, facilitando assim a sua execução.

Defesas orais e avaliação:

- Cada meta terá a sua defesa oral na semana seguinte – **semanas de 13 de abril e 18 de maio** – durante a aula PL e/ou em horário a combinar com o docente da turma PL.
- A avaliação do projeto incidirá principalmente no relatório e código, e estará sempre sujeita à defesa oral em cada meta, com **foco na análise e na discussão dos resultados**.
- O desempenho na defesa influenciará a nota obtida no trabalho implementado. **Sem defesa oral, a nota do trabalho será zero.**

Registo:

**Registe o seu grupo junto do docente da turma PL**, indicando também o participante/pasta escolhido.

# Meta 1 – Análise Temporal e de Frequência

## 09/abril/2026

1. Crie uma estrutura de dados (por exemplo, uma tabela) e guarde as seguintes informações relativas a cada áudio:
  - a. diretório do áudio;
  - b. nome do ficheiro (com extensão);
  - c. participante;
  - d. dígito;
  - e. número da repetição.
2. Desenvolva código para importação dos sinais de áudio e guarde o valor da taxa de amostragem e os sinais na estrutura de dados criada no ponto 1. Nota: pode utilizar a função *audioread* do MATLAB.
3. Reproduza e represente graficamente um exemplo dos sinais importados, identificando o dígito e o número da repetição a que cada um corresponde, e indicando o eixo horizontal com o tempo em segundos, como apresentado no exemplo da Figura 2.

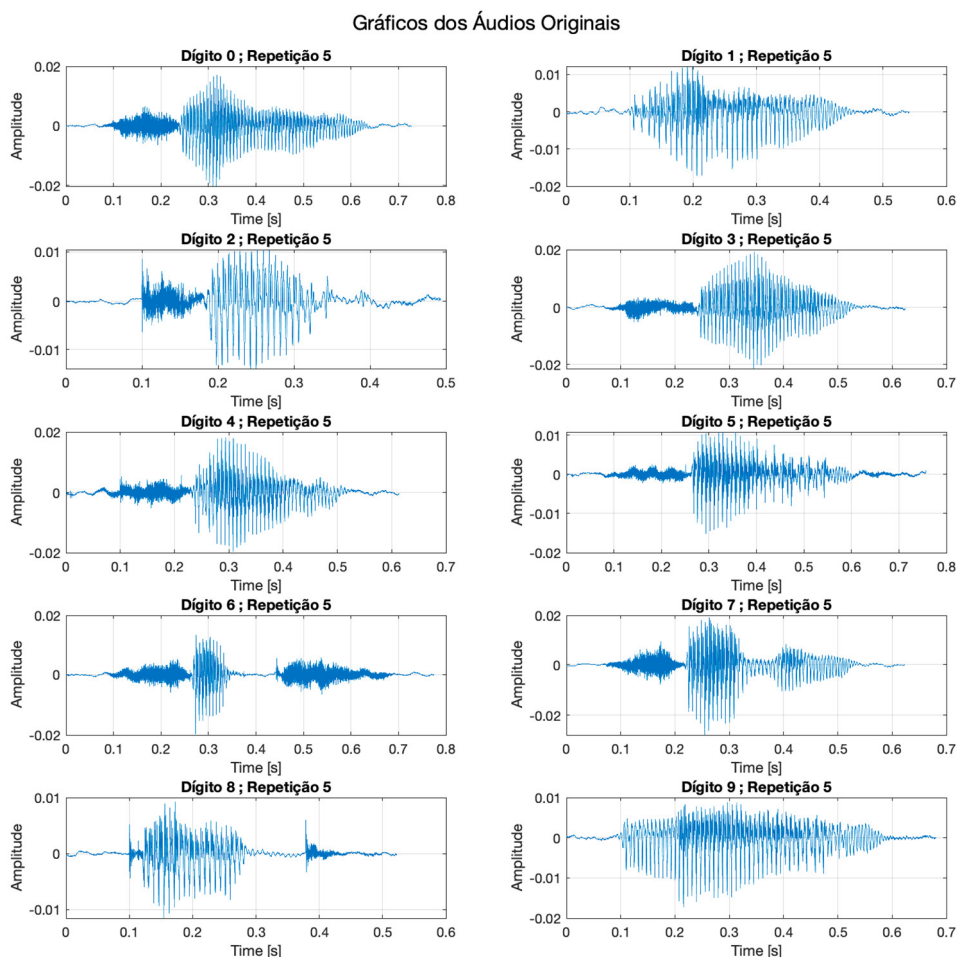


Fig. 2: Sinais de áudio correspondentes aos dígitos de 0 a 9 (repetição 5).

4. Uma estratégia que pode melhorar a diferenciação entre dígitos é fazer uma etapa de pré-processamento para garantir que todos os ficheiros têm a mesma duração e intervalo de amplitude. Para isso deverá implementar os seguintes passos:

- Retirar o “silêncio” inicial dos sinais, de forma a garantir que todos começam exatamente ao mesmo tempo (por exemplo, através do estudo da Energia do sinal ao longo do tempo).
- Normalizar a amplitude com base na amplitude máxima e mínima das amostras. Isto garante que os problemas usuais da gravação do som (por exemplo, a distância da pessoa ao microfone) não interferem na análise.
- Adicionar (ou retirar) silêncio no fim dos ficheiros para garantir que todos têm a mesma duração total.

Guarde os sinais após o pré-processamento na estrutura de dados criada no ponto 1.

5. Para os sinais obtidos após o pré-processamento, repita o ponto 3.
6. Compare os gráficos obtidos para cada dígito nos pontos 3 e 5, e identifique visualmente possíveis características temporais que permitam a diferenciação entre dígitos.
7. Implemente o cálculo de possíveis características (*features*) temporais do sinal que permitam a diferenciação entre dígitos, como energia (total, por intervalos temporais, diferenças entre intervalos, etc.), amplitude (máxima, mínima, razão de amplitudes, desvio padrão, etc.) ou outras medidas temporais que considere relevantes. Deverá efetuar o cálculo para cada áudio de pelo menos 5 características temporais e guardar na estrutura de dados criada no ponto 1.
8. Recorrendo à representação gráfica das características temporais extraídas no ponto anterior, identifique as três características que permitem uma melhor discriminação dos dígitos. Para isto pode utilizar *boxplots*, gráficos 2D, 3D, etc. Alguns exemplos são apresentados na Figura 3.

Indique as características escolhidas, justificando a escolha.

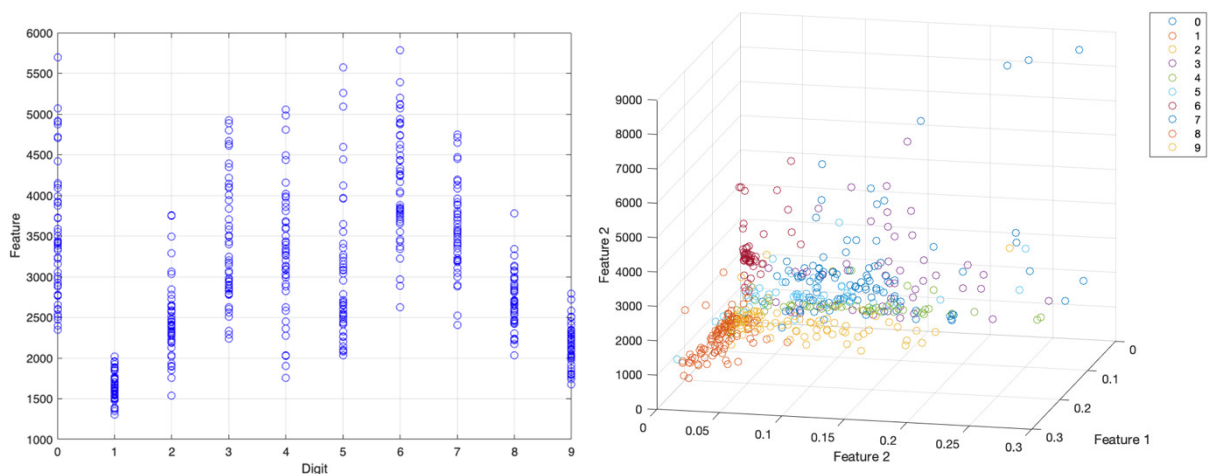


Fig. 3. Representação gráfica das características extraídas.

9. Para cada sinal obtido após o pré-processamento, determine os coeficientes da série complexa de Fourier e guarde na estrutura de dados.
10. Calcule, para cada dígito, o espectro de amplitude mediano, normalizado pelo número de amostras (i.e., equivalente ao módulo dos coeficientes da série complexa de Fourier), e somente para frequências positivas. Calcule também o primeiro quartil (25%) e terceiro quartil (75%). A Figura 4 mostra um exemplo. Compare os gráficos obtidos. Nota: pode utilizar a função *quantile* do MATLAB.

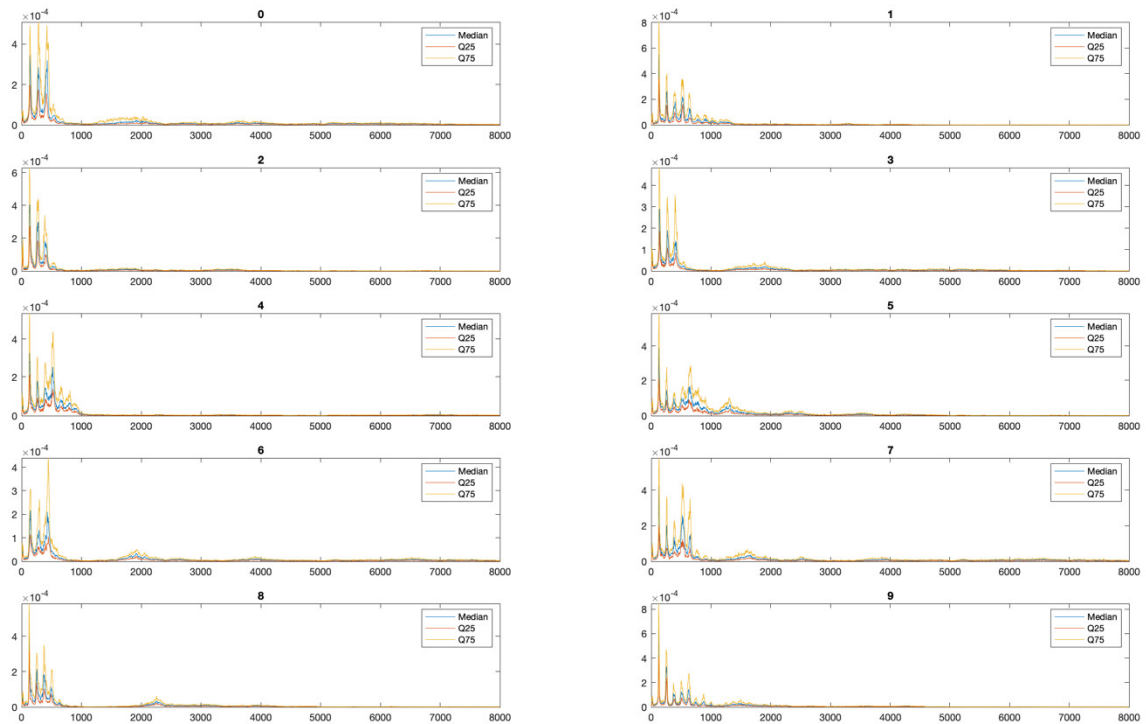


Fig. 4: Espectro de amplitude mediano normalizado, primeiro quartil e terceiro quartil, para os dígitos de 0-9. Para efeitos de uma melhor visualização não é apresentado o eixo de frequências completo.

11. Identifique e implemente o cálculo de possíveis características espectrais que permitam a diferenciação entre dígitos como: máximos espectrais (posição e amplitude), médias espectrais, *spectral edge frequency*, etc. Deverá efetuar o cálculo para cada áudio de peelo menos 5 características espectrais e atualizar a estrutura de dados.
12. Recorrendo à representação gráfica das características espectrais extraídas no ponto anterior, identifique as três características que permitem uma melhor discriminação dos dígitos. Para isto pode utilizar *boxplots*, gráficos 2D, 3D, etc. Indique as características escolhidas, justificando a escolha.
13. Remova da estrutura de dados criada no ponto 1 os sinais de áudio importados e guarde a estrutura de dados num ficheiro `‘.mat’` (passo importante para rapidamente carregar os dados no início da Meta 2).

# Meta 2 – Análise Tempo-Frequência e Classificação

## 14/maio/2026

- Carregue a estrutura de dados (ficheiro `.mat`) guardada no último ponto da meta anterior.
- Calcule, para cada dígito, a STFT, como exemplificado na Figura 5. Use diferentes tipos de janela e diferentes parametrizações, como, por exemplo, tamanho da janela, sobreposição, e número de pontos para cálculo da FFT, e identifique os parâmetros que lhe parecem mais adequados para o objetivo do trabalho. Comente os resultados. Nota: utilize a função `spectrogram` do MATLAB.

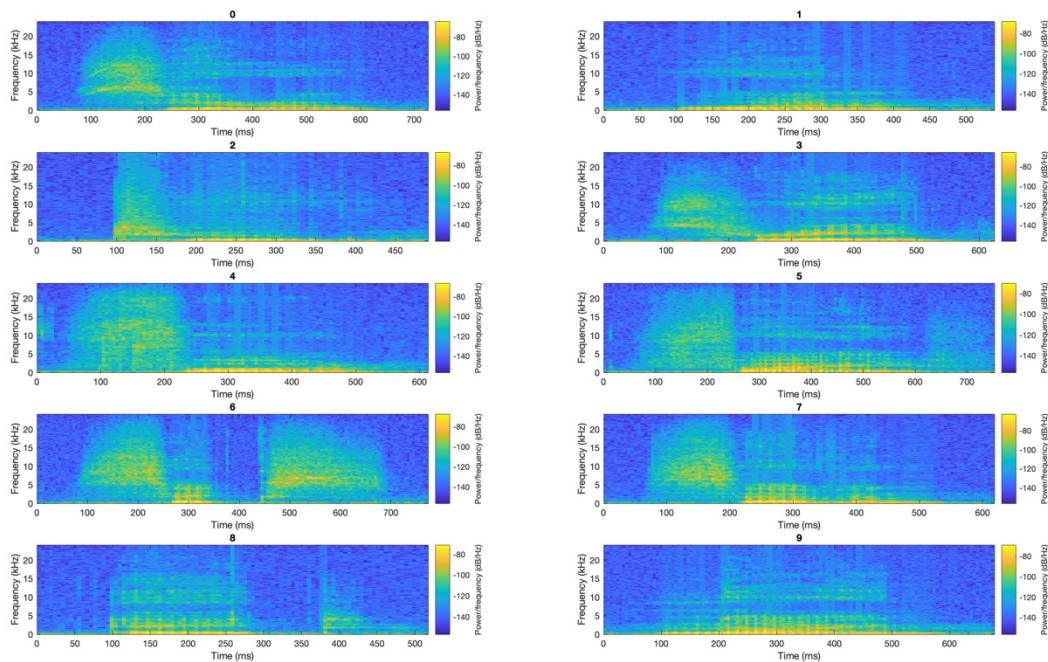


Fig. 5: STFT para os dígitos de 0-9.

- Extraia pelos menos 5 características de tempo-frequência em diferentes janelas tempo-frequência que possam ser mais relevantes para a diferenciação dos dígitos. Deverá atualizar a sua estrutura de dados.
- Recorrendo à representação gráfica das características tempo-frequência extraídas no ponto anterior, identifique as três características que permitem uma melhor discriminação dos dígitos. Para isto pode utilizar `boxplots`, gráficos 2D, 3D, etc. Indique as características escolhidas, justificando a escolha.
- Aplicar a Transformada de Wavelet Discreta (DWT) para obter os coeficientes de detalhe e de aproximação, utilizando um nível de decomposição e o tipo de wavelet que considere apropriado. Obter os valores de energia a partir dos coeficientes e representar os resultados, utilizando gráficos semelhantes aos exemplificados na Figura 3. Compare e discuta os resultados obtidos com a STFT e com a DWT.
- Considerando as melhores características extraídas dos sinais dos pontos 7, 11 e 16, aplique um classificador para a identificação automatizada dos dígitos através das características extraídas. Para o efeito deverá usar um modelo à sua escolha (*Minimum Distance*, um sistema de regras *if/else*, árvores de decisão, etc.) utilizando como *features* as características extraídas e escolhidas até este ponto. Guarde na sua estrutura de dados o(s) resultado(s) da(s) classificação(ões).

20. Compare os dígitos atribuídos a cada áudio a partir do classificador usado no ponto anterior com os dígitos reais e defina a percentagem de acertos (*accuracy*) como:

$$\text{Acertos} = \frac{\text{Dígitos identificados corretamente}}{\text{Total de dígitos}} \times 100$$

Comente os resultados.

21. (Conclusão) Apresente as conclusões finais e uma reflexão crítica fundamentada sobre o trabalho realizado. Aborde os principais desafios, os sucessos alcançados e propostas de melhoria.