

FACULDADE DE TECNOLOGIA DE SÃO PAULO

**PEDRO OTTO DURAN**

Análise de Dados de Uso de Sistemas Computacionais de  
Comunicação Alternativa a Partir de Machine Learning

SÃO PAULO

2025

FACULDADE DE TECNOLOGIA DE SÃO PAULO

**PEDRO OTTO DURAN**

Análise de Dados de Uso de Sistemas Computacionais de  
Comunicação Alternativa a Partir de Machine Learning

Trabalho submetido como exigência parcial  
para a obtenção do Grau de Tecnólogo em  
Análise e Desenvolvimento de Sistemas  
Orientador: Prof. Dr. Marcelo Duduchi

SÃO PAULO

2025

## **AGRADECIMENTOS**

Ao Prof. Dr. Marcelo Duduchi, meu orientador, pela paciência, dedicação e orientação valiosa em todas as etapas deste trabalho. Sua expertise e incentivo foram fundamentais para a conclusão desta pesquisa.

Aos professores do curso, pelo conhecimento compartilhado e pela contribuição à minha formação técnica e humana, que tornaram possível a realização deste projeto.

À minha família, pelo apoio incondicional, compreensão e motivação durante todo o período de desenvolvimento deste trabalho. Seu estímulo foi meu alicerce nos momentos mais desafiadores.

À FATEC-SP, pela estrutura oferecida e pelo ambiente acadêmico que possibilitou o desenvolvimento desta pesquisa e minha formação profissional.

## RESUMO

Sistemas de comunicação alternativa são aqueles os quais tem por objetivo viabilizar a comunicação de pessoas com deficiência. Tais sistemas, requerem um período de adaptação, gerando assim uma “curva de aprendizagem”, que por sua vez define o sucesso de um sistema depois de traçados objetivos e metas. Neste contexto, esta monografia se propõe a analisar um caso específico do uso do Sistema ImagoDiAnaVox, avaliando a melhoria de performance do usuário utilizando técnicas de estatística comuns e criar um modelo de *machine learning* capaz de observar os dados e analisá-los a fim de propor de maneira assertiva em que momento futuro a curva de aprendizagem deverá estar o mais possivelmente próxima de plana, permitindo mensurar a velocidade contra o grau de complexidade de cada sentença, ou parágrafo que o sistema ajudou a criar. Os resultados apresentados mostram as análises estatísticas realizadas e um modelo de Machine Learning criado permite mensurar a velocidade de formação de frases.

**Palavras-chaves:** Comunicação alternativa, Análise Estatística Machine Learning, Aprendizagem

## **ABSTRACT**

Alternative communication systems are those that aim to enable communication for people with disabilities. Such systems require a period of adaptation, thus generating a “learning curve”, which in turn defines the success of a system after objectives and goals have been set. In this context, this monograph proposes to analyze a specific case of the use of the ImagoDiAnaVox System, evaluating the improvement in user performance using common statistical techniques and creating a machine learning model capable of observing the data and analyzing it in order to accurately propose at what future moment the learning curve should be as close to flat as possible, allowing the measurement of speed against the degree of complexity of each sentence or paragraph that the system helped to create. The results presented show the statistical analyses performed and a Machine Learning model created allows the measurement of the speed of sentence formation.

**Keywords:** Alternative Communication, Statistical Analysis Machine Learning, Learning.

## Sumário

1.	INTRODUÇÃO .....	1
1.1	Motivação .....	1
1.2	Objetivo Geral .....	2
1.3	Objetivos Específicos.....	2
2.	Referencial teórico.....	2
2.1	Sistemas de comunicação alternativa.....	2
2.2	Machine Learning.....	4
2.3	<i>Long Short-Term Memory</i> .....	5
3.	Metodologia Científica.....	7
4.	Resultados.....	8
4.1	Descrição e apresentação dos dados.....	8
4.2	Aplicação de Inteligência Artificial .....	12
5.	Conclusão.....	15
	REFERÊNCIAS .....	16
	Apêndice 1 – Código base para execução de todos os outros códigos .....	18
	Apêndice 2 – Funções auxiliares.....	19
	Apêndice 3 – Código Python para extração de gráfico da Figura 3.....	20
	Apêndice 4 – Código fonte para predição de 30 dias.....	21
	Apêndice 5 - Código fonte para predição de 365 dias.....	23

Figura 1 Quantidade de frases/Quantidade de palavras por frase .....	9
Figura 2 Quantidade de frases formadas por mês .....	10
Figura 3:Tempo médio para formação de frases dos diversos tamanhos no tempo de uso do sistema.....	11
Figura 4 Resultados da predição da rede para 30 dias .....	13
Figura 5 Resultados da predição da rede para 365 dias .....	13

## 1. INTRODUÇÃO

Sistemas de comunicação alternativa são aqueles os quais tem por objetivo viabilizar a comunicação de pessoas com deficiência. Por exemplo, o sistema de linguagem de sinais, que se utiliza de insumos físicos (mãos, braços, posições relativas de membros do corpo, etc), ou recursos tecnológicos como ImagoDiAnaVox, que é um sistema computacional que utiliza figuras, anagramas e diacríticos (Capovilla, Duduchi, Macedo, 1998).

O objetivo desses sistemas é, a partir de habilidades preservadas, permitir ao usuário construir frases, expressões ou sentenças, tendo como foco a não utilização de recursos convencionais, para que dessa maneira a pessoa com deficiência adquira a habilidade de se comunicar livremente com outras pessoas, requerendo ou não conhecimento prévio sobre os insumos utilizados no sistema. Em cenários cotidianos existem inúmeras situações possíveis da aplicação desse tipo de sistema, logo, torna-se uma tarefa tão ou até mais complexa avaliar a efetividade desses sistemas tanto com o foco em indivíduos únicos, quanto em maior número.

Todos os sistemas de comunicação alternativa computadorizados requerem um período de adaptação, gerando assim uma “curva de aprendizagem”, que por sua vez define o sucesso de um sistema depois de traçados objetivos e metas.

Neste contexto, esta monografia se propõe a analisar um caso específico do uso do Sistema ImagoDiAnaVox, avaliando a melhoria de performance do usuário utilizando algumas técnicas de estatística comuns. Além disso, criar um modelo de *machine learning* capaz de observar os dados e analisá-los a fim de propor de maneira assertiva em que momento futuro a curva de aprendizagem deverá estar o mais possivelmente próxima de plana, permitindo mensurar a velocidade contra o grau de complexidade de cada sentença, ou parágrafo que o sistema ajudou a criar.

### 1.1 Motivação

Uma parte importante do processo de criação e implantação de um modelo de comunicação alternativa é tentar entender e prever o quão simples será sua adaptação ao usuário e do usuário ao sistema. Logo, com os avanços em machine learning, torna-se possível obter métricas de adaptabilidade no processo de integração das partes, com menos tempo de uso, ampliando assim os dados disponíveis para uma revisão do sistema ou atualização dele.

Tais métricas trazidas de maneira antecipada podem favorecer em muito os usuários, já que é possível, com a devida amostragem de dados analisados, fornecer uma métrica de tempo e efetividade esperadas do sistema para a tomada de decisão de profissionais de saúde.

## 1.2 Objetivo Geral

O objetivo desse projeto é fazer uma análise estatística do uso de um sistema de comunicação alternativa e desenvolver um algoritmo que utilize *machine learning* que seja capaz de construir uma estimativa de tempo para que seja alcançado o melhor fator complexidade da frase / tempo decorrido para sua criação.

## 1.3 Objetivos Específicos

Os objetivos específicos deste projeto são:

- Fazer uma análise dos dados usando estatística descritiva
- Analisar a capacidade de uma rede neural em realizar previsões sobre a curva de aprendizagem de um usuário frente a um sistema de comunicação alternativa.
- Desenvolver um modelo base de inteligência artificial que possa ser treinado com diferentes conjuntos de dados.
- Fornecer previsões similares.

O intuito é que essa solução sirva como uma ferramenta analítica aplicável à avaliação e evolução de sistemas de comunicação alternativa.

## 2. Referencial teórico

O presente trabalho tem como referencial teórico um estudo sobre o conceito e uso de sistemas de comunicação alternativa apresentando os diversos sistemas de comunicação alternativa existentes e seus recursos além de um estudo sobre recursos de machine learning utilizados para construir o sistema de predição presente nesta monografia.

### 2.1 Sistemas de comunicação alternativa

Segundo a *American Speech-Language-Hearing Association Special Interest Division 12*: Comunicação alternativa e aumentativa (CAA) é uma área de pesquisa, clínica e educacional, que envolve estudar e quando necessário compensar de maneiras temporárias ou permanente as ativas limitações de comunicações de indivíduos tanto na compreensão quanto de produção de comunicação escrita ou falada.

As intervenções de CAA e suas tecnologias são partes de um serviço de habilitação ou reabilitação, onde reabilitação se refere a estratégias de intervenção e tecnologias para ajudar indivíduos que estão acometidos por alguma deficiência que

reduz sua competência de comunicação, enquanto a habilitação se refere a estratégias de intervenção e tecnologias para assistir indivíduos que possuem deficiências de desenvolvimento a adquirirem competências de comunicação. (Beukelman, 1998)

Já para Janice Murray (2009), CAA inclui quaisquer métodos de comunicação que suplementam ou substituem os métodos convencionais de fala/escrita quanto estes não suportam as necessidades do indivíduo.

Como dito por Sally Millar (1998), em transcrição livre e resumida: A Comunicação Alternativa e Aumentativa (CAA) abrange quaisquer métodos de comunicação que complementam ou substituem a fala e a escrita convencionais, especialmente quando estas habilidades estão comprometidas. Indivíduos com diversas deficiências frequentemente dependem de técnicas especializadas de CAA, pois a fala pode não ser seu principal meio de interação. O propósito fundamental da CAA é maximizar as capacidades comunicativas existentes do indivíduo, mitigando ou compensando as funções prejudicadas. Com os avanços tecnológicos recentes e uma crescente conscientização sobre a diversidade de opções de comunicação disponíveis para pessoas com ampla gama de deficiências, observa-se um potencial significativo para aprimorar o nível de comunicação de um número cada vez maior de usuários.

Embora o uso de aspectos da CAA seja comum no dia a dia (por exemplo, gestos em ambientes ruidosos ou ao se comunicar em língua estrangeira), para muitas pessoas, a dependência da CAA é constante. Sua aplicação é vasta, abrangendo desde ambientes domésticos e instituições de ensino (creches, escolas, faculdades e ensino superior) até hospitais, unidades de terapia intensiva, centros de dia e residências assistidas, onde a comunicação é essencial.

A CAA possui um potencial transformador na qualidade de vida dos usuários. Conforme expressado por Macdonald (1994), "a alegria de poder escrever novamente e manter contato com velhos amigos [...] é tremenda. Ter a segurança de saber que sempre terei alguma forma de conversar com as pessoas mais próximas a mim é maravilhoso demais para descrever." Essa capacidade de comunicação amplia a participação social e a independência, como destacado por McFadden (1995): "[...] para mim e para outros como eu, ser capaz de se comunicar nos insere na sociedade. Isso nos permite ter uma voz. [...] Para mim, ter um "*Liberator*" mudou completamente a minha vida." Além disso, a CAA potencializa o acesso a oportunidades de aprendizado e educação: "Usar o '*BIGmack*' tem sido muito benéfico. Permitiu que ele participasse de várias atividades em sala de aula e ajudou a fortalecer sua compreensão de símbolos, adicionando fala a eles" (professor de crianças com dificuldades de aprendizagem profundas e complexas).

A Comunicação Aumentativa e Alternativa (CAA) engloba um conjunto de métodos, estratégias e tecnologias desenvolvidos para ampliar ou substituir a fala

e/ou a escrita de indivíduos com deficiências na comunicação funcional. Esses sistemas visam proporcionar meios eficazes de expressão, utilizando recursos como imagens, textos, sons, gestos, entre outros, para facilitar a interação social e a participação ativa em diversos contextos.

A CAA é aplicável a uma ampla gama de indivíduos, incluindo pessoas surdas ou com deficiência auditiva, que podem se beneficiar de linguagens de sinais ou sistemas baseados em símbolos visuais. Pessoas com paralisia cerebral, por exemplo, podem utilizar dispositivos computacionais equipados com pranchas de comunicação ou softwares especializados para formar palavras e expressar pensamentos. Um exemplo notável é o físico Stephen Hawking, que utilizava um sistema de CAA para se comunicar eficientemente, demonstrando que, com o suporte adequado, é possível superar barreiras significativas na comunicação.

A Comunicação Alternativa e Aumentativa (CAA) representa um pilar fundamental para a promoção da inclusão social, educacional e profissional de indivíduos que apresentam deficiências significativas na comunicação. Ao transcender as limitações impostas pela fala e escrita convencionais, a CAA oferece um leque de estratégias e ferramentas que capacitam esses sujeitos à expressar suas necessidades, desejos, pensamentos e sentimentos. Esse empoderamento comunicativo não apenas facilita a participação ativa em diversos contextos da vida cotidiana, mas também lhes assegura o pleno exercício de seus direitos, consolidando sua presença e contribuição na sociedade.

É crucial compreender que a competência comunicativa em CAA vai muito além da mera manipulação de ferramentas tecnológicas ou analógicas. Conforme enfatizado por Light e McNaughton (2014). Ela engloba a capacidade de interação social efetiva, ou seja, a habilidade de usar os recursos da CAA para engajar-se em diálogos significativos, negociar significados e construir relacionamentos. Nesse sentido, os Sistemas de CAA — sejam eles de baixa tecnologia (como pranchas de comunicação) ou de alta tecnologia (como comunicadores eletrônicos e aplicativos) — são indispensáveis para garantir a inclusão social plena de indivíduos com dificuldades de comunicação, permitindo-lhes romper barreiras e participar ativamente em todos os aspectos da vida comunitária, como bem pontuam Beukelman e Mirenda (2013). A implementação desses sistemas não se restringe à substituição da fala, mas visa ampliar as possibilidades comunicativas, adaptando-se às necessidades individuais e potencializando a autonomia do usuário.

## **2.2 Machine Learning**

É considerado fácil explicar para uma pessoa o que caracteriza a diferença entre um carro normal e um de corrida, mostrando apenas fotos de exemplo de ambos, dessa maneira formulando regras claras que definem o que caracteriza um carro de corrida (Janiesch, 2021). De maneira similar, ao contrário de programar

diretamente essas regras, o processo de *Machine Learning* aparenta aprender sozinho as relações importantes e padrões com base em observação. (Bishop, 2006)

Considerado um subconjunto da Inteligência Artificial (IA), o aprendizado de máquina (ML) exhibe o "aprendizado" experiencial associado à inteligência humana, além de possuir a capacidade de aprender e aprimorar suas análises por meio do uso de algoritmos computacionais. Esses algoritmos utilizam grandes conjuntos de entradas e saídas de dados para reconhecer padrões e "aprender" efetivamente, a fim de treinar a máquina a tomar recomendações ou tomar decisões autônomas. Após repetições e modificações suficientes do algoritmo, a máquina torna-se capaz de receber uma entrada e prever uma saída. As saídas são então comparadas com um conjunto de resultados conhecidos para avaliar a precisão do algoritmo, que é então ajustado iterativamente para aperfeiçoar a capacidade de prever resultados futuros (Helm, 2020).

Computadores "aprendem" em *machine learning* ao aprimorar seu desempenho em tarefas através da "experiência". Na prática, "experiência" geralmente significa o ajuste a dados. Portanto, não há uma fronteira clara entre *machine learning* e abordagens estatísticas. De fato, se uma dada metodologia é considerada "machine learning" ou "estatística" muitas vezes reflete sua história tanto quanto diferenças genuínas, e muitos algoritmos (por exemplo, LASSO – "*least absolute shrinkage and selection operator*", regressão passo a passo) podem ou não ser considerados *machine learning* dependendo de quem você pergunta. Ainda assim, apesar das similaridades metodológicas, *machine learning* é filosoficamente e praticamente distinguível. Correndo o risco de uma (considerável) simplificação excessiva, *machine learning* geralmente enfatiza a precisão preditiva em detrimento da inferência baseada em hipóteses, geralmente focando em conjuntos de dados grandes e de alta dimensão (ou seja, com muitas covariáveis) (Chen, 2018).

Como já dito, o processo de aprendizado das máquinas, se baseia em expor um algoritmo computacional a um conjunto de dados com o objetivo de deixar o algoritmo "aprender" os padrões dos dados, sem a necessidade de programar essas regras, e dessa maneira se tornar capaz de discernir e reconhecer padrões dentro dos dados, sendo capaz assim futuramente de prever, classificar ou separar informações que são posteriormente enviadas para esse algoritmo.

### **2.3 Long Short-Term Memory**

Conforme Lipton et al. (2016), as redes *Long Short-Term Memory* (LSTM) são eficazes em modelar sequências de dados temporais, capturando dependências de longo prazo que outras arquiteturas não conseguem.

Redes LSTM são um tipo especial de RNN, capazes de aprender dependências de longo prazo. Elas foram introduzidas inicialmente por *Hochreiter* e

*Schmidhuber (1997)* e funcionam muito bem em uma grande variedade de problemas, sendo amplamente utilizadas atualmente.

Segundo Pedro César (2024), “A *Long Short-Term Memory* (LSTM) é um tipo de rede neural recorrente (RNN) projetada para resolver o problema das dependências de longo prazo em dados sequenciais. Introduzidas por Hochreiter e Schmidhuber em 1997, as LSTM são eficazes em tarefas que exigem a memorização de informações por longos períodos, como tradução automática, previsão de séries temporais e reconhecimento de fala.

A principal inovação das LSTM é sua capacidade de manter e atualizar informações por longos períodos por meio de três portas principais: porta de entrada, porta de esquecimento e porta de saída. Essas portas regulam o fluxo de informações dentro da célula de memória, permitindo que a rede lembre e esqueça seletivamente informações importantes”.

Dentro desse escopo de *Machine Learning*, as redes neurais recorrentes (RNN) são naturalmente indicadas para análise de séries temporais, mas sofrem com o problema do desvanecimento do gradiente em sequências muito longas. Para resolver isso, foram propostas as redes LSTM (*Long Short-Term Memory*), que incorporam mecanismos de “portas” capazes de controlar o fluxo de informação e mitigar o esquecimento gradual, permitindo a retenção de dependências de longo prazo.

Essa capacidade torna as LSTM particularmente apropriadas para prever a curva de aprendizagem de usuários em sistemas de comunicação alternativa, pois permitem modelar com precisão as variações de desempenho ao longo do tempo, mesmo diante de longos históricos de uso, e dessa maneira gerar previsões sobre a performance do usuário junto ao sistema.

### 3. Metodologia Científica

Os dados utilizados neste estudo foram coletados ao longo de 25 meses, oriundos da interação de um único indivíduo com paralisia cerebral, cuja condição impossibilitava a comunicação por meios convencionais. As informações foram registradas durante o uso contínuo de um sistema de comunicação alternativa, baseado na seleção de símbolos e palavras por meio de uma interface visual.

Antes da análise, os dados passaram por um processo de limpeza e padronização. Primeiramente, foi implementado um algoritmo personalizado responsável pela remoção de entradas em branco e pela eliminação de repetições consecutivas de frases, a fim de preservar a integridade e a diversidade do conteúdo da amostra. Em seguida, foi aplicada normalização dos valores numéricos utilizando a técnica Min-Max Scaling, através do `MinMaxScaler()` da biblioteca Scikit-learn, garantindo que todos os dados fossem transformados para um intervalo entre 0 e 1.

A análise exploratória inicial segmentou as frases com base na quantidade de palavras (de uma a oito). Posteriormente, os dados foram reorganizados em séries temporais com agregação diária, permitindo o cálculo da média do tempo de formação de frases por dia, por quantidade de palavras. Essa estrutura de dados possibilitou visualizar a evolução da curva de aprendizado do usuário ao longo do tempo.

Para a etapa de modelagem preditiva, foi implementada uma rede neural do tipo *Long Short-Term Memory* (LSTM), uma arquitetura especialmente indicada para dados sequenciais e séries temporais, por sua capacidade de capturar dependências de longo prazo. O modelo foi construído utilizando a biblioteca Keras, com as seguintes camadas:

- 5 camadas LSTM com 25, 50, 100, 50 e 25 neurônios, todas com função de ativação ReLU e `return_sequences=True`, exceto a última;
- Uma camada densa intermediária com 5 neurônios e ativação ReLU;
- Uma camada densa de saída com 1 neurônio, também com ativação ReLU, responsável pela previsão do tempo estimado de formação da frase.

O modelo foi compilado com o otimizador Adam e a função de perda Erro Quadrático Médio (MSE). O treinamento foi realizado por 50 épocas, com batch size de 16, utilizando dados organizados em janelas temporais ( $n\_timesteps$ ) como entrada (X), e os tempos médios como saída (y).

Ao final, o modelo foi capaz de estimar, com base no histórico de uso, o tempo necessário para a formação de frases de diferentes tamanhos, permitindo uma representação aproximada da curva de aprendizagem individual observada durante o período analisado.

## 4. Resultados

A seguir são apresentados o cenário da avaliação e os resultados obtidos a partir das análises realizadas e a aplicação das técnicas de Inteligência Artificial. Os códigos usados para obter os resultados a seguir estão em Apêndices. No apêndice 1 está o código usado como base para a execução dos demais códigos. No apêndice 2 estão as funções auxiliares usadas. Os códigos dos demais apêndices foram usados para apresentação dos resultados que serão citados a medida que os resultados obtidos forem apresentados.

### 4.1 Descrição e apresentação dos dados

Os dados neste trabalho foram coletados ao longo de 25 meses de uso contínuo do sistema ImagoDiAnaVox por um indivíduo com paralisia cerebral, que não possui a capacidade de se comunicar por meios convencionais. A base de dados for exportada em uma estrutura tabular onde os campos são:

- Data: Registro da data de formação da frase (formato DD/MM/AAAA).
- Segundos: Tempo gasto (em segundos) para construção da frase.
- Palavras: Número total de palavras na frase (varia de 1 a 8).
- PLV1 a PLV8: Campos que armazenam cada palavra da frase (posições não preenchidas são marcadas com "-").

Exemplo de registro real:

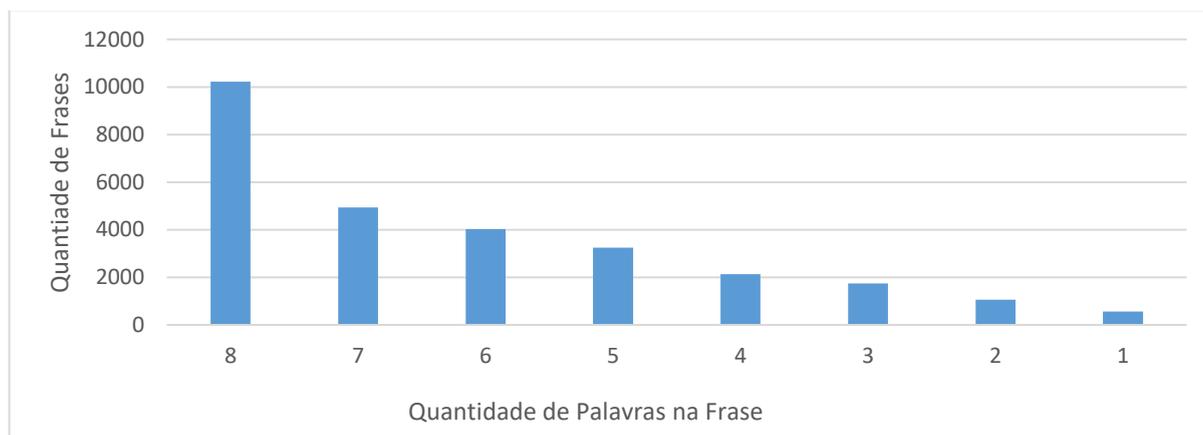
Data	Segundos	Palavras	PLV1	PLV2	PLV3	PLV4	PLV5	PLV6	PLV7	PLV8
14/02/1996	739	6	FAZ	MUITO	TEMPO (HORA)	VIR	AQUI	MINHA CASA	-	-
14/02/1996	237	8	FAZ	MUITO	TEMPO (HORA)	NAO	VIR	AQUI	EM	MINHA CASA
14/02/1996	290	5	JA	ESTAR	COM	MUITA	SAUDADE	-	-	-

A estrutura de dados permitiu diferentes formas de agrupamento e visualização dos dados, com destaque para a visualização referente a distribuição de número de palavras por frase, que se observa na Figura 1, observando os seguintes dados percentuais.

- 8 Palavras: 36,60%
- 7 Palavras: 17,67%
- 6 Palavras: 14,40%
- 5 Palavras: 11,64%

- 4 Palavras: 7,63%
- 3 Palavras: 6,23%
- 2 Palavras: 3,79%
- 1 Palavra: 2,04%

Figura 1 Quantidade de frases/Quantidade de palavras por frase



### Resultado da pesquisa

Observa-se, portanto, uma distribuição desigual do número de palavras por frase, com predominância de frases mais longas. Esta base de dados serviu como insumo para a análise estatística e preditiva desenvolvida nos tópicos seguintes, oferecendo um panorama robusto do comportamento do usuário frente a construção de frases com maior grau de complexidade.

Como evidenciado na Figura 2, o sistema foi intensamente utilizado durante o período de coleta de dados, embora com variação significativa na frequência mensal. Nota-se, em particular, que abril de 1995 registrou o menor volume de atividade, contabilizando apenas 477 entradas de frases – valor substancialmente inferior à média observada nos demais meses

Figura 2 Quantidade de frases formadas por mês

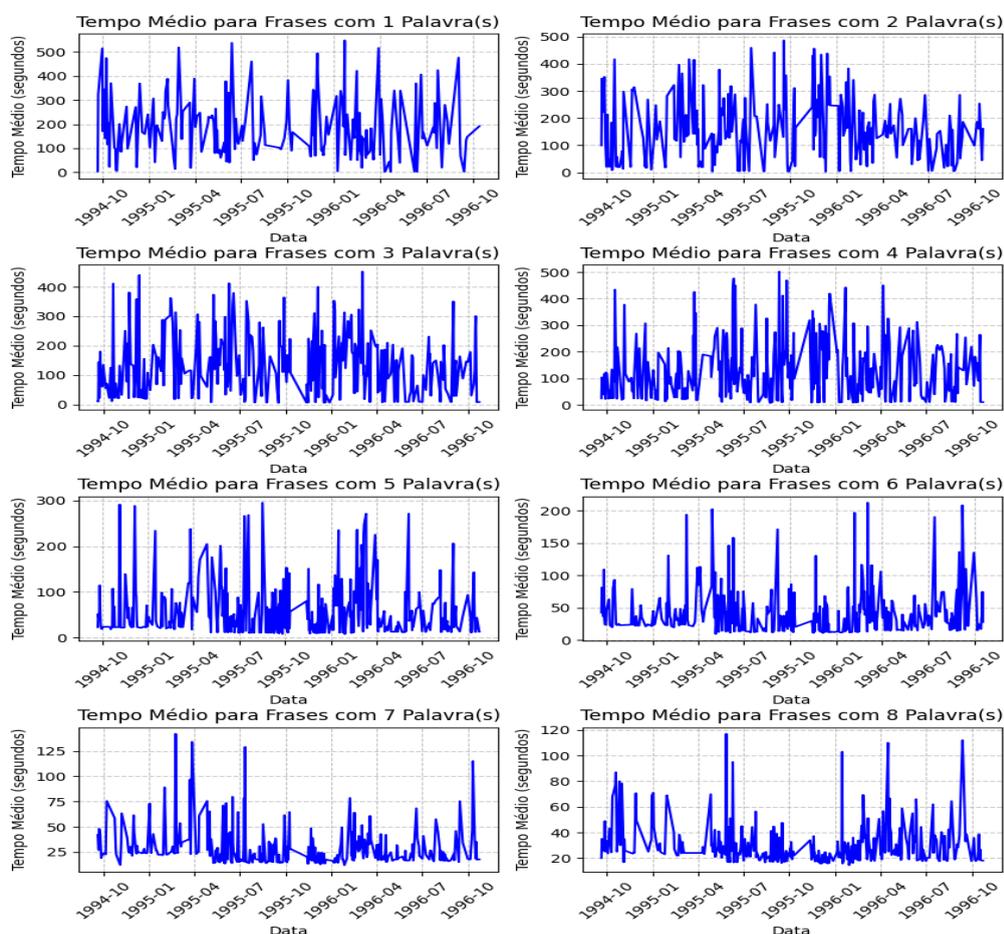


O cenário utilizado para este trabalho consiste em uma base de dados do ImagoDiAnaVox que possui uma abrangência de 25 meses, que foi utilizada por uma pessoa com paralisia cerebral, sendo assim um bom exemplo de uso real da tecnologia e plataforma. Para a análise, os dados foram normalizados primeiramente removendo repetições seguidas da mesma frase, já que o sistema onde foram coletados os dados permitia a repetição da última frase, gerando assim distorção na análise dos dados. O próximo passo foi realizar uma filtragem por quartil para reduzir anormalidades na massa de dados.

Por se tratar de um sistema que possui até oito palavras por frase, foi realizada a análise de tempo médio das frases em relação ao tempo para cada comprimento de frase, ou seja, tempo médio de formação da frase / data de criação, assim possibilitando observar a evolução da velocidade de criação de frases de tamanhos variados ao longo do tempo.

A Figura 3 mostra o tempo de formação de frases dos diversos tamanhos durante o período de uso do sistema. Ela foi criada a partir da extração de dados da base utilizando-se de um script em Python presente no Apêndice 3.

Figura 3: Tempo médio para formação de frases dos diversos tamanhos no tempo de uso do sistema



### Resultado da pesquisa

É possível avaliar que até as frases com 5 palavras existe uma tendência a diminuição do tempo do tempo médio, mesmo que pouca. Já para as frases com 6 palavras encontra-se uma linearidade, demonstrando que durante o tempo de uso esse foi o tamanho de frase ao qual o usuário melhor se adaptou.

Já para as frases com 7 e 8 palavras é possível avaliar inicialmente um aumento do tempo de frase, porém com uma ascendência a partir do meio da linha do tempo, sendo assim é possível concluir que existiu uma curva de aprendizado, que apesar de inicialmente ter dificultado o uso do sistema, conforme foi decorrendo o tempo, facilitou o seu uso.

Estes pontos acima destacados são o histórico e a análise do ponto de vista de o que o sistema já proveu ao usuário. Estes dados nos auxiliam no entendimento

da usabilidade do sistema e da sua capacidade de fornecer um meio de comunicação alternativa ao usuário.

## 4.2 Aplicação de Inteligência Artificial

Com os dados entendidos e avaliados, é possível criar tanto um modelo matemático, quanto uma inteligência artificial para prover previsões da curva de aprendizagem, e demonstrar através de gráfico uma quantidade de tempo na qual se pode esperar melhoras significativas, e em qual momento poderemos esperar uma linearidade onde o tempo de frase será mais ou menos equalizado, assim tendo, uma perspectiva de efetividade do uso do sistema de comunicação alternativa.

Para isso será usada uma rede neural do tipo LTSM (Long Short-Term Memory) que é adequada a esse tipo de aplicação por conta de sua característica de maior retenção de informações, e capacidade de gerar dados de maneira progressiva predizendo os resultados futuros para o usuário.

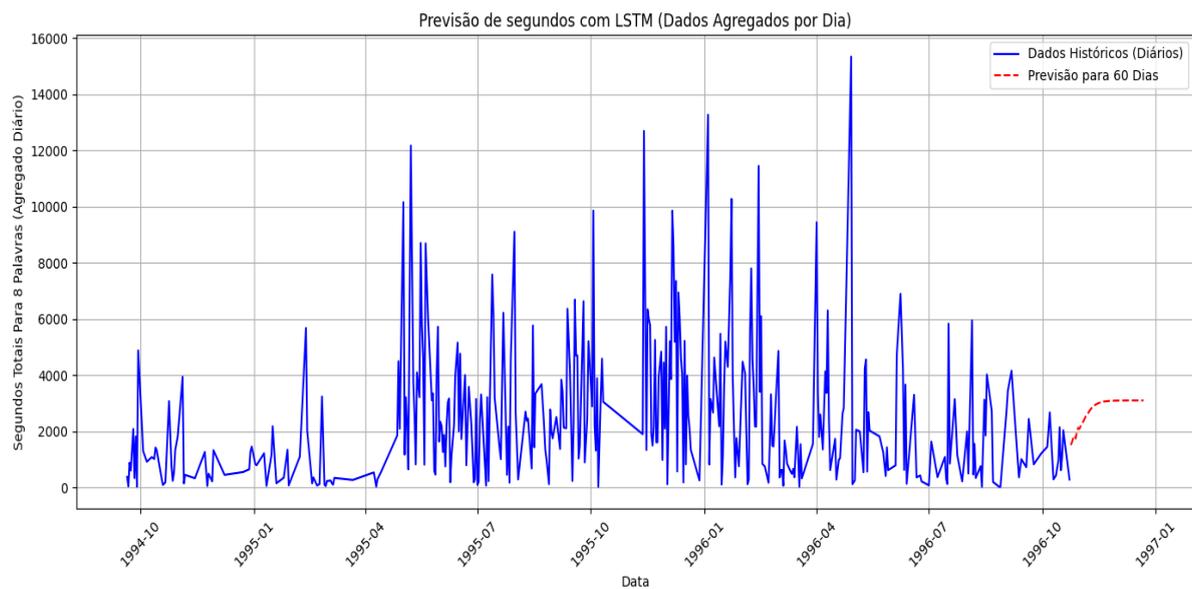
Esta modelagem preditiva fundamenta-se em princípios estabelecidos por Hochreiter e Schmidhuber (1997), que demonstraram a superioridade das redes LSTM (*Long Short-Term Memory*) para análise de sequências temporais complexas. Como observado por Gers et al. (2000), "a arquitetura LSTM é particularmente eficaz para problemas que envolvem dependências de longo prazo em séries temporais", característica essencial para modelar curvas de aprendizagem.

O treinamento foi realizado em cima do cenário de frases que possuem 8 palavras, tendo como alvo o cenário mais complexo do processo de comunicação que a ferramenta disponibiliza, refletindo melhor o seu uso cotidiano.

Como destacado por Light et al. (2019), modelos preditivos em CAA requerem validação em múltiplos usuários. As principais limitações identificadas foram:

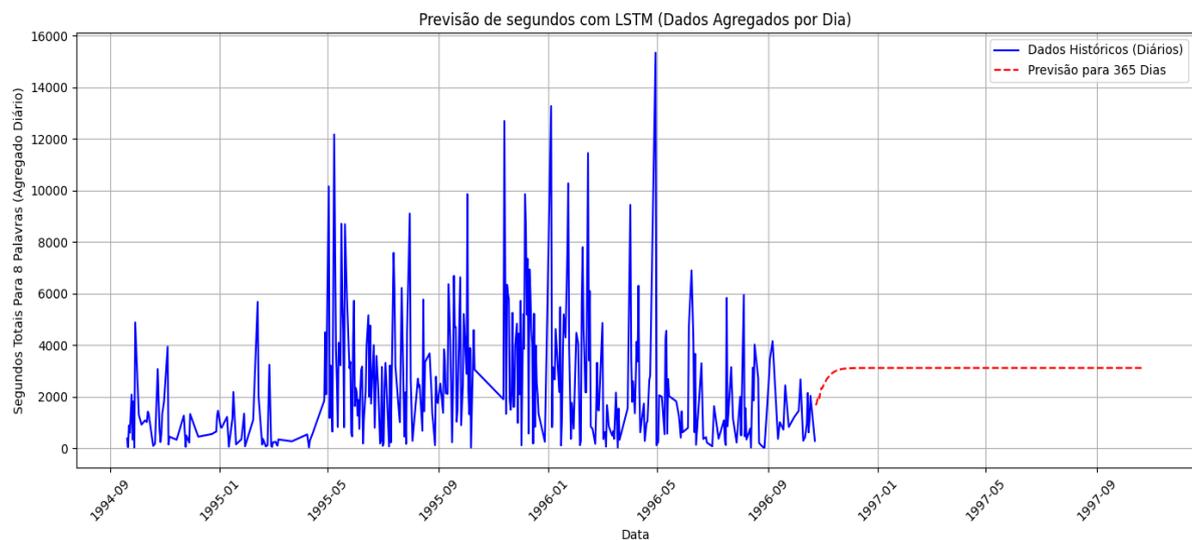
- Base única de usuário (viés individual)
- Janela temporal limitada (25 meses)
- Necessidade de ampliação do *dataset* para melhor generalização

Figura 4 Resultados da predição da rede para 60 dias



Resultado da pesquisa

Figura 5 Resultados da predição da rede para 365 dias



Resultado da pesquisa

As Figuras 4 e 5, feitas a partir do código do apêndice 4, ilustram a integração dos dados históricos observados (especificamente, as frases com oito palavras) com as projeções do modelo LSTM. Essas projeções se estabilizam nos limites de 2000 a 3000 segundos a partir de 30 dias de predição, mostram a grande capacidade do modelo em prever a curva de aprendizagem do usuário. Contudo, é fundamental destacar que a acurácia preditiva do modelo sofre uma degradação progressiva a partir do 30º dia, uma limitação diretamente atribuída à quantidade restrita de dados

históricos disponíveis. Essa tendência de perda de precisão foi consistentemente observada e confirmada ao testar a mesma arquitetura LSTM em horizontes temporais mais amplos, como 60 e 365 dias. Nesses cenários estendidos, foram reveladas inconsistências significativas e um aumento acentuado no erro preditivo conforme o período de projeção se amplia, o que, por sua vez, compromete a aplicabilidade prática do modelo para previsões de longo prazo

Uma análise detalhada dos gráficos (Figuras 4 e 5) mostra que a aparente linearidade nas previsões representa, na verdade, um artefato das limitações inerentes à arquitetura da rede neural. Especificamente, o comportamento errático que emerge após os 30 dias evidencia claramente os limites operacionais do modelo quando aplicado a horizontes temporais mais extensos. Esse fenômeno está alinhado com a literatura especializada, que reconhece este padrão de decaimento da acurácia como característico de modelos LSTM em cenários de predição de longo prazo (Yu et al., 2019).

Os resultados obtidos sugerem que, embora o modelo demonstre capacidade satisfatória para capturar tendências e padrões em curtos intervalos de tempo (até 30 dias), sua eficácia diminui progressivamente quando aplicado a horizontes mais amplos. Esta limitação reforça a importância de implementar mecanismos de retreinamento periódico do modelo, garantindo assim que as previsões mantenham sua relevância e acurácia ao longo do tempo. Além disso, os achados destacam a necessidade de cautela na interpretação de resultados preditivos para períodos superiores a um mês, sugerindo que esta janela temporal representa o limite prático para aplicações confiáveis do modelo em seu estado atual.

O presente estudo optou por não considerar a disposição ou as distintas dificuldades de acesso das figuras disponíveis no sistema, as quais estão organizadas em diversas categorias. Assumiu-se que quaisquer variações decorrentes dessas diferenças seriam diversificadas e, por conseguinte, diluídas ao longo das distintas categorias de acesso consideradas na análise. Essa premissa foi adotada visando manter o foco principal na análise da curva de aprendizagem e na eficácia do modelo LSTM em um contexto mais amplo de interação do usuário.

Esta análise comparativa não apenas delimita o escopo de aplicabilidade do modelo desenvolvido, mas também fornece valiosos insights para futuras melhorias na arquitetura, particularmente no que diz respeito ao tratamento de dependências temporais de longo prazo em sistemas de comunicação alternativa. Os resultados obtidos abrem caminho para investigações futuras sobre técnicas de aumento de memória em redes neurais recorrentes e estratégias de atualização contínua dos modelos preditivos.

## 5. Conclusão

Depois das análises prévias e das análises dos resultados das redes, é possível entender que o processo de predição é possível e assertivo até certo ponto, porém o número limitado de dados de 25 meses torna o processo de treinamento da rede mais complexo devido à falta de dados. Apesar de limitada a capacidade de predição foi possível sim criar um modelo que seja capaz de fazer predições mais assertivas e por um período maior.

Outro ponto de limitação é o fato de a base de dados ser de apenas um indivíduo, isso torna o cenário mais impreciso em relação ao uso da ferramenta, uma vez que a análise de indivíduos isolados não retrata de maneira satisfatória as capacidades do sistema, já que existem inúmeros fatores humanos inerentes ao processo de uso da ferramenta.

Por fim, pode-se concluir que o modelo obteve sucesso em estimar em média o tempo da formação das frases entre 2000 e 3000 segundos para um indivíduo apenas, porém por um período de 30 dias somente, sendo necessários mais dados e de um número maior de indivíduos para aumentar a acurácia e extensão de tempo possível de geração dos dados

## REFERÊNCIAS

BEUKELMAN, D. R.; MIRENDA, P. *Augmentative and alternative communication: supporting children and adults with complex communication needs*. Baltimore: Paul H. Brookes Publishing Co., 2013.

BEUKELMAN, D. R. et al. *Augmentative and alternative communication*. Baltimore: Paul H. Brookes, 1998.

BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. New York: Springer, 2006.

CAPOVILLA, F. C.; DUDUCHI, M.; MACEDO, E. C. Sistema de multimídia para comunicação picto-silábica: análise de seis meses de uso domiciliar de ImagoVox por mulher com paralisia cerebral. *Ciência Cognitiva*, v. 2, n. 1, p. 21-112, 1998.

CHEN, Z.; LIU, B. *Lifelong machine learning*. San Rafael: Morgan & Claypool Publishers, 2018.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to forget: continual prediction with LSTM. *Neural Computation*, v. 12, n. 10, p. 2451-2471, 1 out. 2000.

HELM, J. M. et al. Machine learning and artificial intelligence: definitions, applications, and future directions. *Current Reviews in Musculoskeletal Medicine*, v. 13, p. 69-76, 2020.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735-1780, nov. 1997.

JANIESCH, C.; ZSCHECH, P.; HEINRICH, K. Machine learning and deep learning. *Electronic Markets*, v. 31, n. 3, p. 685-695, 2021.

KONG, Y. et al. Unlocking the power of LSTM for long term time series forecasting. *Annals of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, p. 11968-11976, Pennsylvania, fev. 2025.

LIGHT, J.; MCNAUGHTON, D. Communicative competence for individuals who require augmentative and alternative communication: a new definition for a new era of communication? *Augmentative and Alternative Communication*, v. 30, n. 1, p. 1-18, 2014.

LIGHT, J. et al. Designing effective AAC displays for individuals with developmental disabilities. *Augmentative and Alternative Communication*, v. 35, n. 4, p. 217-229, 2019.

MILLAR, S. et al. What is augmentative and alternative communication? An introduction. *Augmentative Communication in Practice*, v. 1, n. 1, p. 3-12, 1998.

MURRAY, J.; GOLDBART, J. Augmentative and alternative communication: a review of current issues. *Pediatrics and Child Health*, v. 19, n. 10, p. 464-468, 2009.

YU, Y. et al. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, v. 31, n. 7, p. 1235-1270, jul. 2019.

## Apêndice 1 – Código base para execução de todos os outros códigos

```
df = remover_repeticoes_consecutivas(df)
print('tipo analise')
df['DT'] = pd.to_datetime(df['DT'])
df['SEGS'] = pd.to_numeric(df['SEGS'], errors='coerce')
df['NPLV'] = pd.to_numeric(df['NPLV'])
df['SEG_PLV'] = pd.to_numeric(df['SEG_PLV'])
start_date = df['DT'].min()
end_date = df['DT'].max()
df['DIA'] = df['DT'].dt.to_period('D').apply(lambda r: r.start_time)
df['semana'] = df['DT'].dt.to_period('W').apply(lambda r: r.start_time)
```

## Apêndice 2 – Funções auxiliares

```

import matplotlib.pyplot as plt
import pandas as pd

def ajustar_margem(contagem, margem_inferior_pct=0.10, margem_superior_pct=0.80):
    margem_inferior = margem_inferior_pct * (contagem.max() - contagem.min())
    margem_superior = margem_superior_pct * (contagem.max() - contagem.min())
    plt.ylim(contagem.min() - margem_inferior, contagem.max() + margem_superior)

def grafico_linhas(vl1, vl2, title, xlabel, ylabel):
    plt.figure(figsize=(10, 6))
    plt.plot(vl1, vl2, marker="o", linestyle='-', color='b')
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.xticks(rotation=45)
    plt.grid(True, linestyle='--', alpha=0.7)
    plt.tight_layout()

def grafico_barra(vl1, vl2, title, xlabel, ylabel):
    plt.figure(figsize=(10, 6))
    plt.bar(vl1, vl2, color='b', width=0.6)
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.grid(True, linestyle='--', alpha=0.7, axis='y')
    plt.xticks(vl1, rotation=45)
    plt.tight_layout()

def remover_outliers_por_quartil(df):
    Q1 = df['SEGS'].quantile(0.25)
    Q3 = df['SEGS'].quantile(0.75)
    IQR = Q3 - Q1

    limite_inferior = Q1 - 1.5 * IQR
    limite_superior = Q3 + 1.5 * IQR

    df_filtrado = df[(df['SEGS'] >= limite_inferior) & (df['SEGS'] <= limite_superior)]
    return df_filtrado

def remover_repeticoes_consecutivas(df):
    cols_plv = ['PLV1', 'PLV2', 'PLV3', 'PLV4', 'PLV5', 'PLV6', 'PLV7', 'PLV8']
    condicao = df[cols_plv] != df[cols_plv].shift()
    df_filtrado = df[condicao.any(axis=1)].reset_index(drop=True)
    return df_filtrado

```

### Apêndice 3 – Código Python para extração de gráfico da Figura 3

```
df = remover_repeticoes_consecutivas(df)
df_filtrado = df[df["NPLV"].between(1, 8)]
fig, axes = plt.subplots(4, 2, figsize=(10, 6))
axes = axes.flatten()
for i, num_palavras in enumerate(range(1, 9)):
    print(f'num_palavras {num_palavras} | i {i}')
    df_num_palavras = df_filtrado[df_filtrado["NPLV"] == i + 1]
    df_num_palavras = remover_outliers_por_quartil(df_num_palavras)
    tempo_medio_por_dia = df_num_palavras.groupby('DIA')['SEGS'].mean()
    axes[i].plot(tempo_medio_por_dia.index, tempo_medio_por_dia.values, marker="x", linestyle='-',
color='b')
    axes[i].set_title(f'Tempo Médio para Frases com {num_palavras} Palavra(s)')
    axes[i].set_xlabel('Data')
    axes[i].set_ylabel('Tempo Médio (segundos)')
    axes[i].grid(True, linestyle='--', alpha=0.7)
    axes[i].tick_params(axis='x', rotation=45)
plt.subplots_adjust(hspace=0.6)
```

## Apêndice 4 – Código fonte para predição de 30 dias

```

import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
import pyodbc
import matplotlib.pyplot as plt

def remover_repeticoes_consecutivas(df):
    cols_plv = ['PLV1', 'PLV2', 'PLV3', 'PLV4', 'PLV5', 'PLV6', 'PLV7', 'PLV8']
    condicao = df[cols_plv] != df[cols_plv].shift()
    df_filtrado = df[condicao.any(axis=1)].reset_index(drop=True)
    return df_filtrado

conn = pyodbc.connect(f'DRIVER={{ODBC Driver 17 for SQL
Server}};SERVER={server};DATABASE={database};UID={username};PWD={password}')
df = pd.read_sql(query, conn)
conn.close()
df = remover_repeticoes_consecutivas(df)
df['DT'] = pd.to_datetime(df['DT'])
df['SEGS'] = pd.to_numeric(df['SEGS'], errors='coerce')
df['NPLV'] = pd.to_numeric(df['NPLV'])
df['SEG_PLV'] = pd.to_numeric(df['SEG_PLV'])
df['DIA'] = df['DT'].dt.to_period('D').apply(lambda r: r.start_time)
df['semana'] = df['DT'].dt.to_period('W').apply(lambda r: r.start_time)
df = df[df['NPLV'] == 8]
df_daily = df.groupby('DIA')['SEGS'].sum().reset_index() # ou .mean()
target = 'SEGS'
dados = df_daily[target].dropna().values.reshape(-1, 1)
scaler = MinMaxScaler()
dados_normalizados = scaler.fit_transform(dados)
def create_sequences(data, n_steps):
    X, y = [], []
    for i in range(len(data) - n_steps):
        X.append(data[i:i+n_steps])
        y.append(data[i+n_steps])
    return np.array(X), np.array(y)

n_timesteps = 7 # Janela de 7 dias (ajuste conforme sazonalidade)
X, y = create_sequences(dados_normalizados, n_timesteps)
X = X.reshape((X.shape[0], X.shape[1], 1))

# =====
# 2. Modelagem LSTM
# =====
model = Sequential()
model.add(LSTM(25, activation='relu', return_sequences=True, input_shape=(n_timesteps, 1)))
model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(n_timesteps, 1)))
model.add(LSTM(100, activation='relu', return_sequences=True, input_shape=(n_timesteps, 1)))

```

```

model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(n_timesteps, 1)))
model.add(LSTM(25, activation='relu', return_sequences=False, input_shape=(n_timesteps, 1)))
model.add(Dense(5, activation='relu'))
model.add(Dense(1, activation='relu'))
model.compile(optimizer='adam', loss='mse')
history = model.fit(X, y, epochs=50, batch_size=16, verbose=1)

```

```
# =====
```

```
# 3. Previsão Futura
```

```
# =====
```

```

def forecast_future(model, last_sequence, n_future):
    future_predictions = []
    current_seq = last_sequence.copy()
    for _ in range(n_future):
        next_pred = model.predict(current_seq.reshape(1, n_timesteps, 1), verbose=0)[0, 0]
        future_predictions.append(next_pred)
        current_seq = np.roll(current_seq, -1)
        current_seq[-1] = next_pred
    return np.array(future_predictions)

```

```
last_window = X[-1]
```

```
n_future = 30 # numero de dias a prever
```

```
future_predictions = forecast_future(model, last_window, n_future)
```

```
future_predictions = scaler.inverse_transform(future_predictions.reshape(-1, 1))
```

```
dados_originais = scaler.inverse_transform(dados_normalizados)
```

```
# =====
```

```
# 4. Plotagem com Datas Diárias
```

```
# =====
```

```
last_date = df_daily['DIA'].iloc[-1]
```

```
future_dates = pd.date_range(start=last_date, periods=n_future + 1, freq='D')[1:]
```

```
plt.figure(figsize=(14, 6))
```

```
plt.plot(df_daily['DIA'], dados_originais, label='Dados Históricos (Diários)', color='blue')
```

```
plt.plot(future_dates, future_predictions, label=f'Previsão para {n_future} Dias', color='red', linestyle='--')
```

```
plt.xlabel('Data')
```

```
plt.ylabel('Segundos Totais Para 8 Palavras (Agregado Diário)')
```

```
plt.title('Previsão de segundos com LSTM (Dados Agregados por Dia)')
```

```
plt.legend()
```

```
plt.grid(True)
```

```
plt.xticks(rotation=45)
```

```
plt.tight_layout()
```

```
plt.show()
```

## Apêndice 5 - Código fonte para predição de 365 dias

```

import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
import pyodbc
import matplotlib.pyplot as plt

def remover_repeticoes_consecutivas(df):
    cols_plv = ['PLV1', 'PLV2', 'PLV3', 'PLV4', 'PLV5', 'PLV6', 'PLV7', 'PLV8']
    condicao = df[cols_plv] != df[cols_plv].shift()
    df_filtrado = df[condicao.any(axis=1)].reset_index(drop=True)
    return df_filtrado

conn = pyodbc.connect(f'DRIVER={{ODBC Driver 17 for SQL
Server}};SERVER={server};DATABASE={database};UID={username};PWD={password}')
df = pd.read_sql(query, conn)
conn.close()
df = remover_repeticoes_consecutivas(df)
df['DT'] = pd.to_datetime(df['DT'])
df['SEGS'] = pd.to_numeric(df['SEGS'], errors='coerce')
df['NPLV'] = pd.to_numeric(df['NPLV'])
df['SEG_PLV'] = pd.to_numeric(df['SEG_PLV'])
df['DIA'] = df['DT'].dt.to_period('D').apply(lambda r: r.start_time)
df['semana'] = df['DT'].dt.to_period('W').apply(lambda r: r.start_time)
df = df[df['NPLV'] == 8]
df_daily = df.groupby('DIA')['SEGS'].sum().reset_index() # ou .mean()
target = 'SEGS'
dados = df_daily[target].dropna().values.reshape(-1, 1)
scaler = MinMaxScaler()
dados_normalizados = scaler.fit_transform(dados)
def create_sequences(data, n_steps):
    X, y = [], []
    for i in range(len(data) - n_steps):
        X.append(data[i:i+n_steps])
        y.append(data[i+n_steps])
    return np.array(X), np.array(y)

n_timesteps = 7 # Janela de 7 dias (ajuste conforme sazonalidade)
X, y = create_sequences(dados_normalizados, n_timesteps)
X = X.reshape((X.shape[0], X.shape[1], 1))

# =====
# 2. Modelagem LSTM
# =====
model = Sequential()
model.add(LSTM(25, activation='relu', return_sequences=True, input_shape=(n_timesteps, 1)))
model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(n_timesteps, 1)))
model.add(LSTM(100, activation='relu', return_sequences=True, input_shape=(n_timesteps, 1)))

```

```

model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(n_timesteps, 1)))
model.add(LSTM(25, activation='relu', return_sequences=False, input_shape=(n_timesteps, 1)))
model.add(Dense(5, activation='relu'))
model.add(Dense(1, activation='relu'))
model.compile(optimizer='adam', loss='mse')
history = model.fit(X, y, epochs=50, batch_size=16, verbose=1)

# =====
# 3. Previsão Futura
# =====
def forecast_future(model, last_sequence, n_future):
    future_predictions = []
    current_seq = last_sequence.copy()
    for _ in range(n_future):
        next_pred = model.predict(current_seq.reshape(1, n_timesteps, 1), verbose=0)[0, 0]
        future_predictions.append(next_pred)
        current_seq = np.roll(current_seq, -1)
        current_seq[-1] = next_pred
    return np.array(future_predictions)

last_window = X[-1]
n_future = 365 # numero de dias a prever
future_predictions = forecast_future(model, last_window, n_future)

future_predictions = scaler.inverse_transform(future_predictions.reshape(-1, 1))
dados_originais = scaler.inverse_transform(dados_normalizados)

# =====
# 4. Plotagem com Datas Diárias
# =====
last_date = df_daily['DIA'].iloc[-1]
future_dates = pd.date_range(start=last_date, periods=n_future + 1, freq='D')[1:]

plt.figure(figsize=(14, 6))
plt.plot(df_daily['DIA'], dados_originais, label='Dados Históricos (Diários)', color='blue')
plt.plot(future_dates, future_predictions, label=f'Previsão para {n_future} Dias', color='red', linestyle='--')
plt.xlabel('Data')
plt.ylabel('Segundos Totais Para 8 Palavras (Agregado Diário)')
plt.title('Previsão de segundos com LSTM (Dados Agregados por Dia)')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```