

# INTELIGÊNCIA ARTIFICIAL NO PROCESSO CRIATIVO MUSICAL

*UM ESTUDO DE CASO DE GERAÇÃO  
DE MÚSICA COM REDES NEURAIS*

Projeto Final de Graduação em Engenharia de  
Produção Universidade Federal Fluminense (UFF).

Autor: JOÃO PEDRO SANTOS MURAD

Orientador: JOSÉ KIMIO ANDO, D. Sc.



# Música como Produto

## 🎵 Contexto

- Música é um produto digital de consumo massivo
  - Alta demanda do *streaming* na indústria musical
  - 69% da receita global (\$ 19,3 Bi)



# Música como Produto

## Problema

- Como produzir música de forma eficiente para atender a demanda crescente?



# Música como Produto

## Solução Investigada

- Uso da Inteligência Artificial (IA)
  - Auxiliar e aprimorar a composição e o arranjo musical
  - Acelerar a produção



# IA como Ferramenta Criativa

Anos 1950

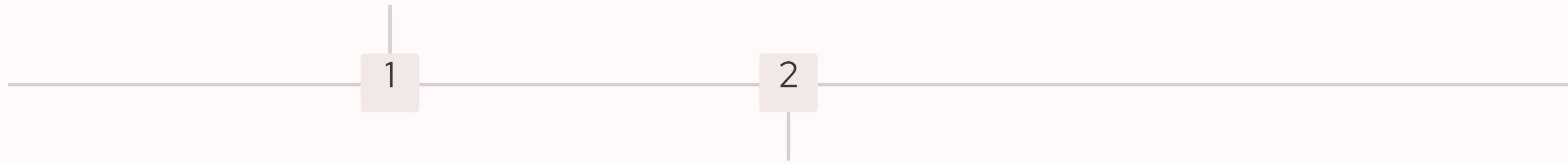
Alan Turing,  
Manchester Mark II

1

# IA como Ferramenta Criativa

Anos 1950

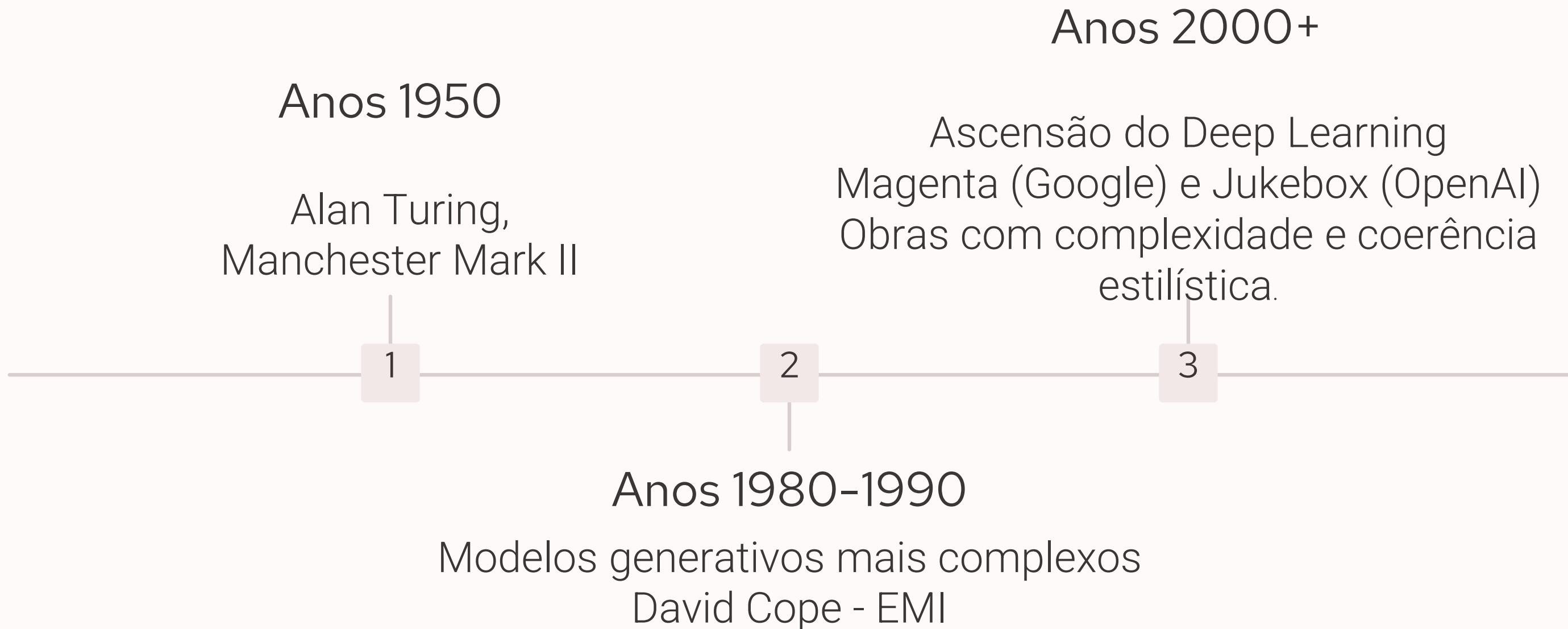
Alan Turing,  
Manchester Mark II



Anos 1980-1990

Modelos generativos mais complexos  
David Cope - EMI

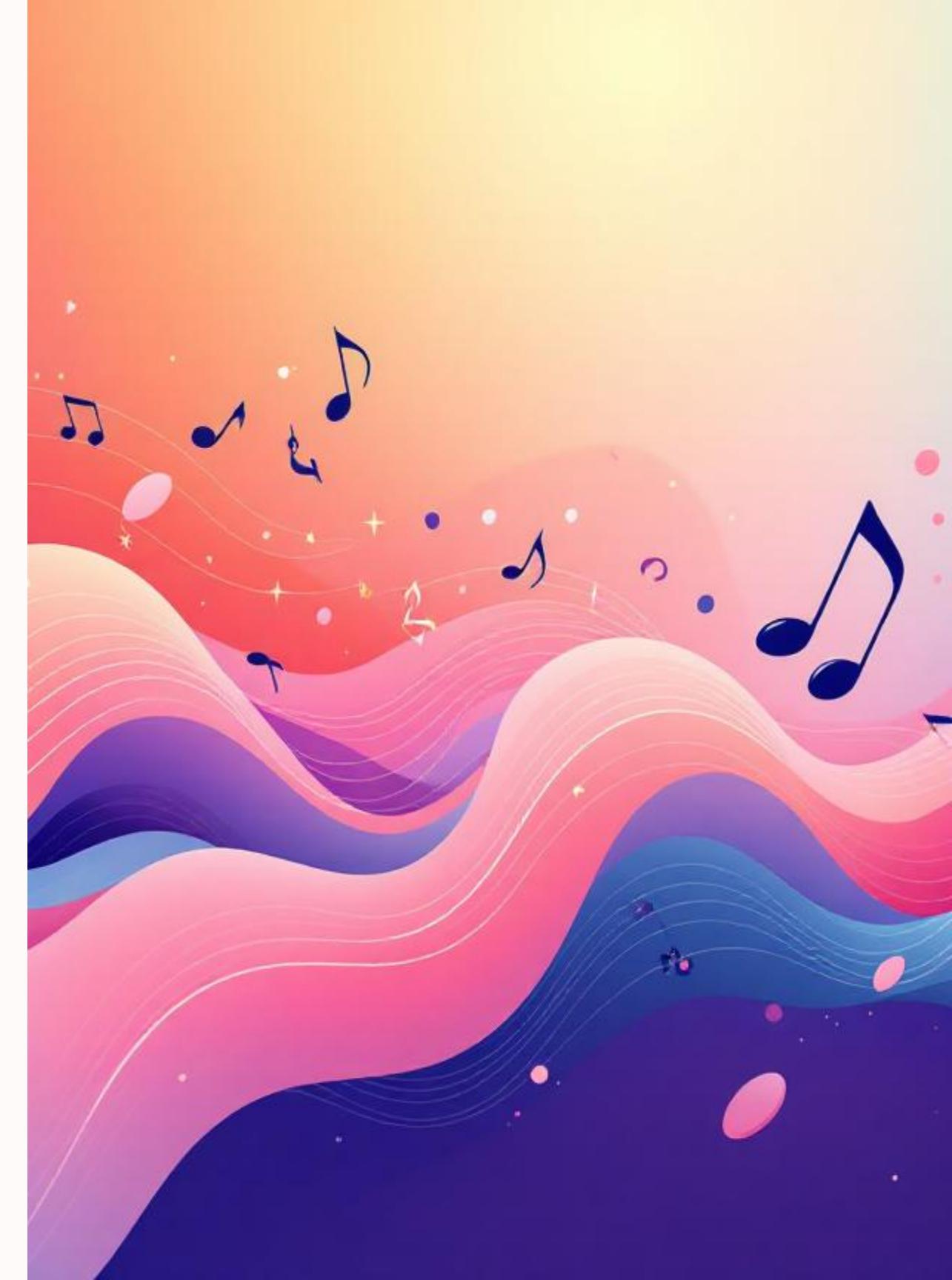
# IA como Ferramenta Criativa



# Bifurcação Tecnológica: Recorrência vs. Atenção

## *LSTM (Long Short-Term Memory)*

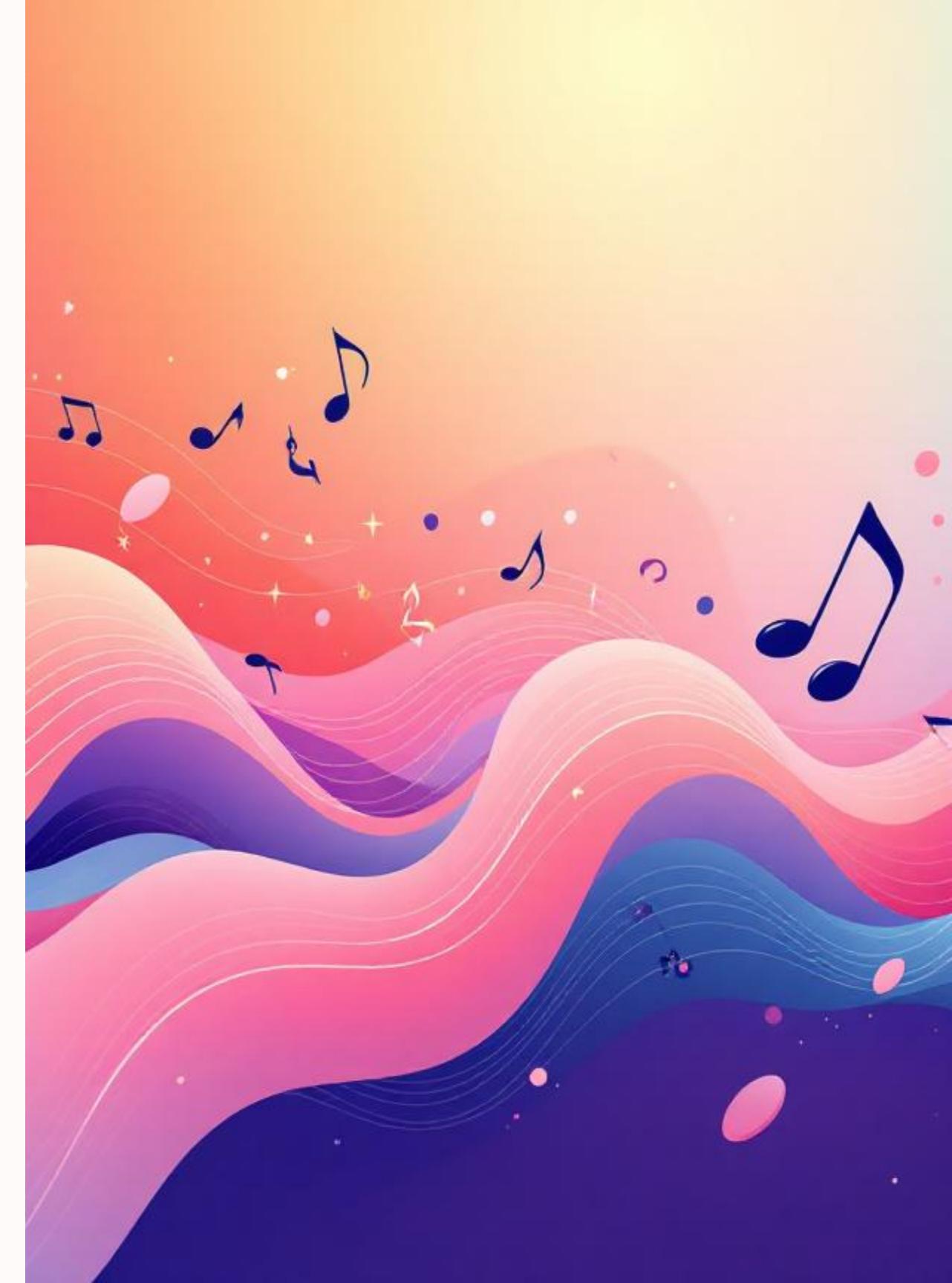
- Recorrente:
  - Processa informação sequencial;
  - Mantém memória interna
  - Boa para curto prazo



# Bifurcação Tecnológica: Recorrência vs. Atenção

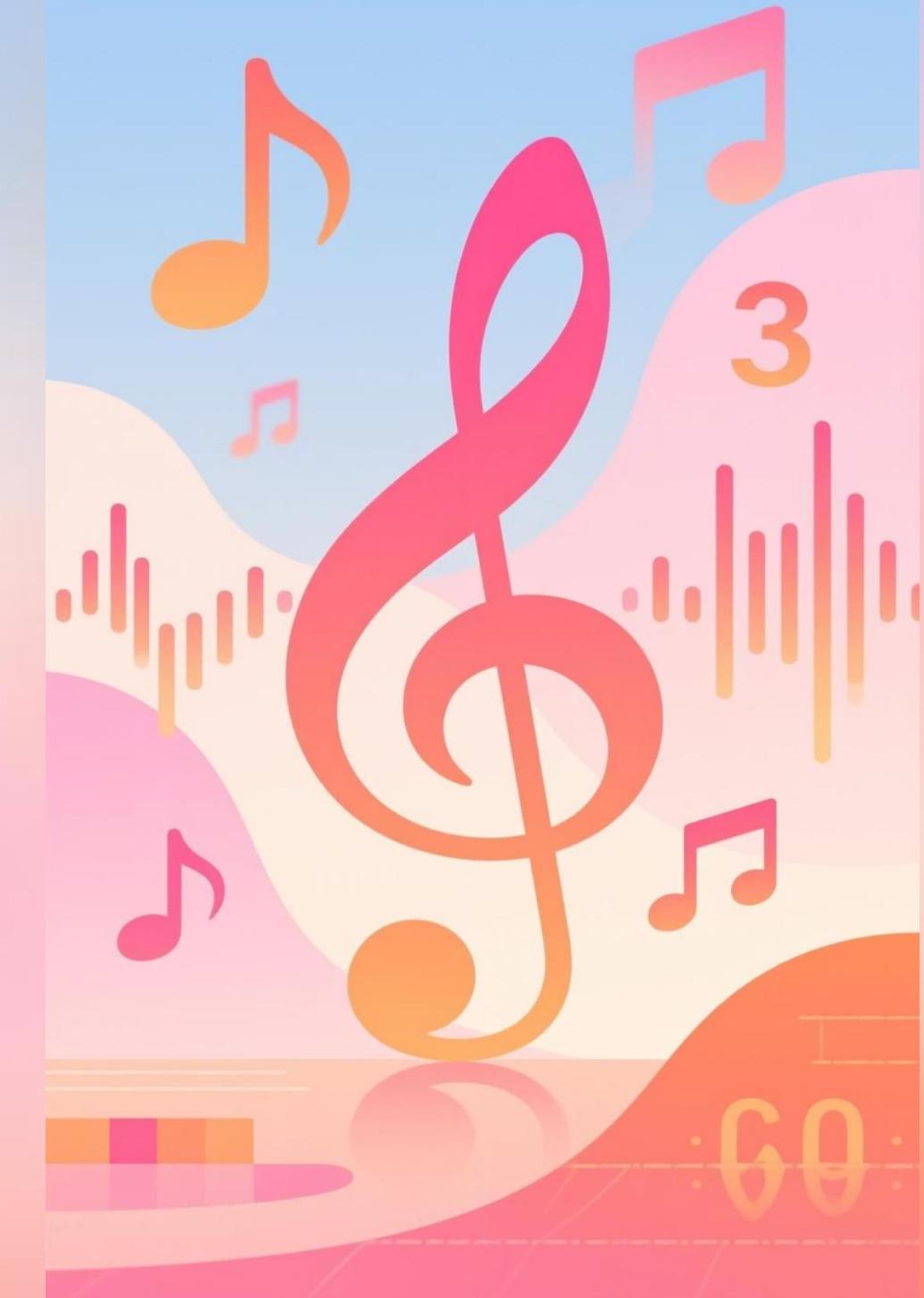
## *Transformer*

- Atenção:
  - Autoatenção para ponderar importância dos elementos simultaneamente;
  - Superior para estruturas longas



# Como a IA "Entende" a Música

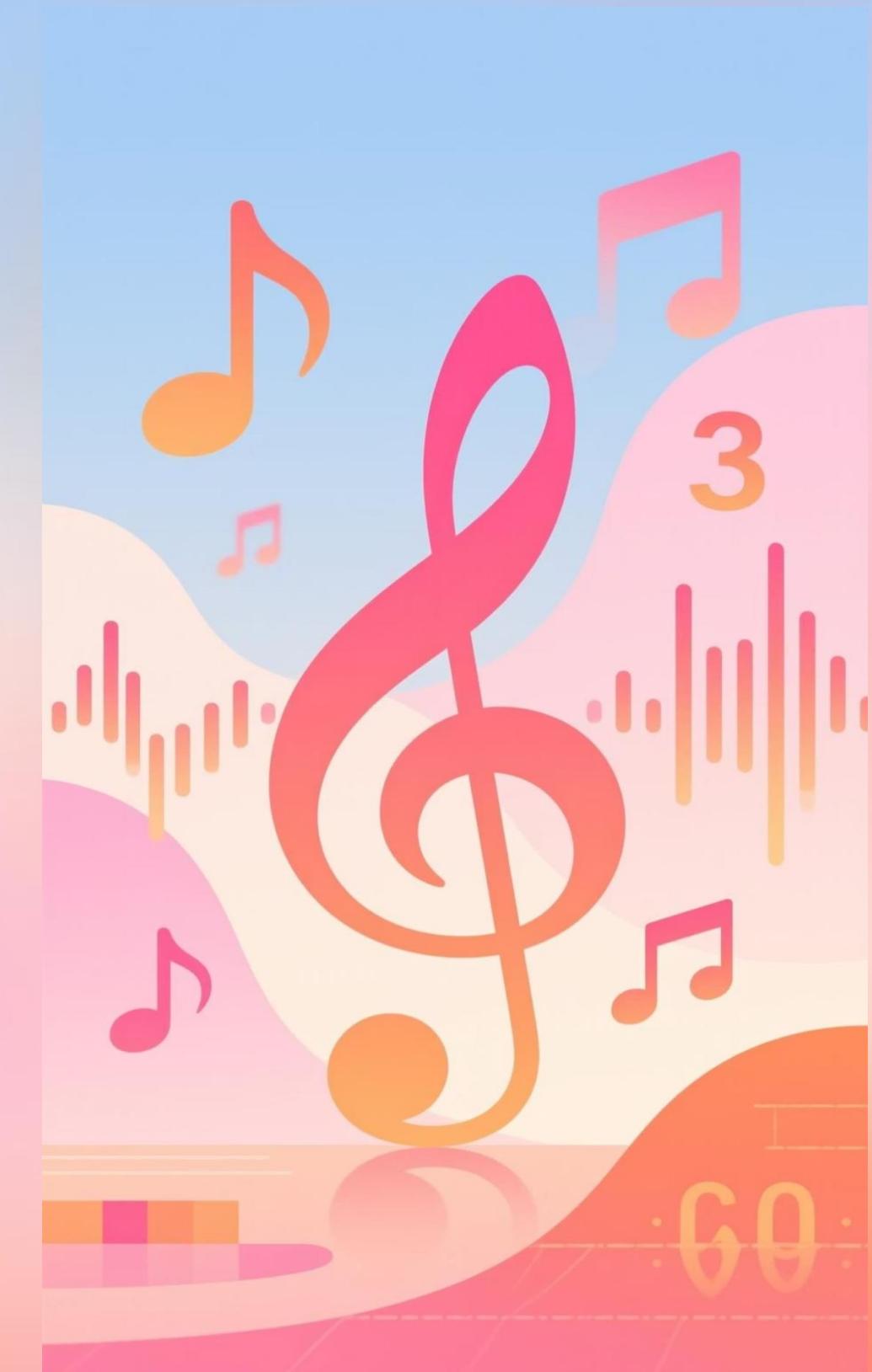
- A música precisa ser **“traduzida”** em dados numéricos
- **Abordagens**
  - Representação Simbólica
  - Representação Parametrizada



# Como a IA "Entende" a Música

## *Abordagem 1: Representação Simbólica*

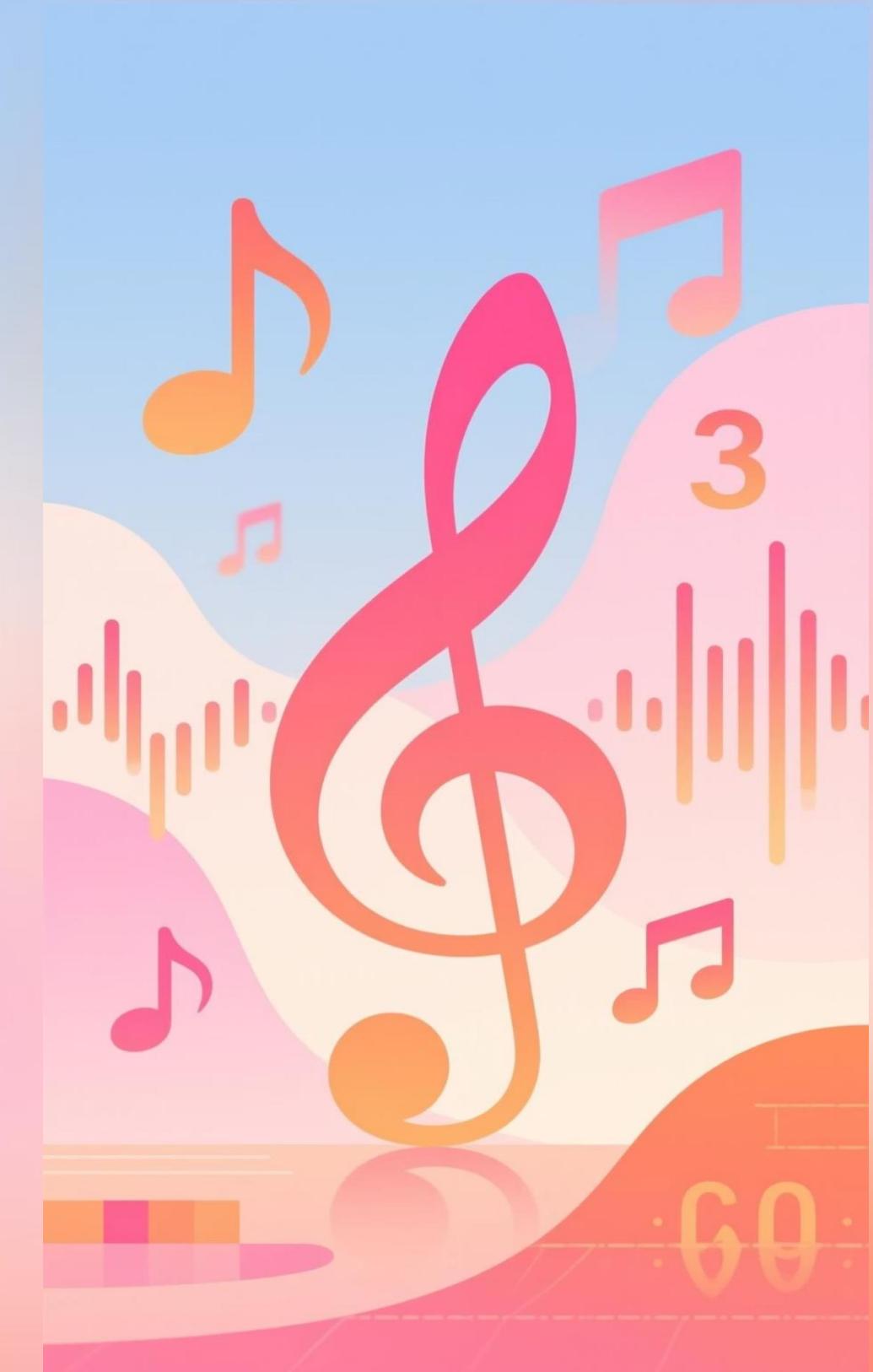
- Trata notas e acordes como "palavras".
  - Simples de Implementar
  - Perde informações de ritmo e duração.
  - Biblioteca `music21`.



# Como a IA "Entende" a Música

## *Abordagem 2: Representação Parametrizada*

- Trata notas como um vetor com diversos parâmetros.
  - Mantém informações rítmicas e temporais
  - Mais complexo para modelar e processar
  - Biblioteca pretty-midi



# Metodologia do Estudo de Caso



*Datasets*

- Chopin: composições de um único compositor.
- MAESTRO: 170 horas de música. Vários compositores e estilos.



# Metodologia do Estudo de Caso



## *Experimento (8 Cenários)*

- Rodada 1: 4 testes  
(LSTM/Transformer vs.  
Chopin/MAESTRO)
- Rodada 2: 4 testes com modelos  
maiores e ajustados.



# Metodologia do Estudo de Caso



## Avaliação

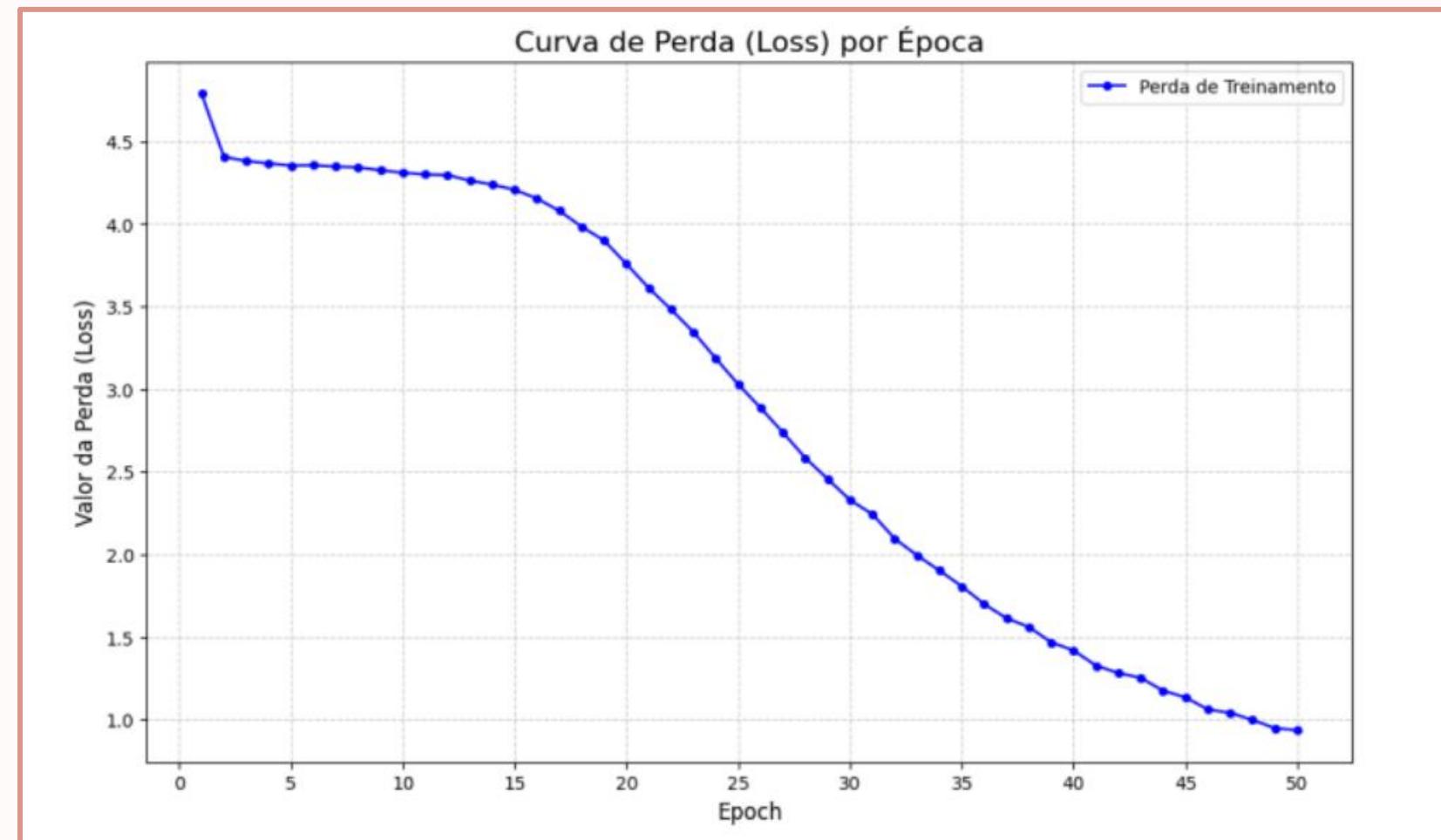
- Quantitativa: Análise do tempo de treino/recursos de hardware
- Qualitativa: 5 ouvintes avaliando. Matriz GUT (Melodia, Harmonia, Ritmo)



# Análise Quantitativa: Custo Computacional

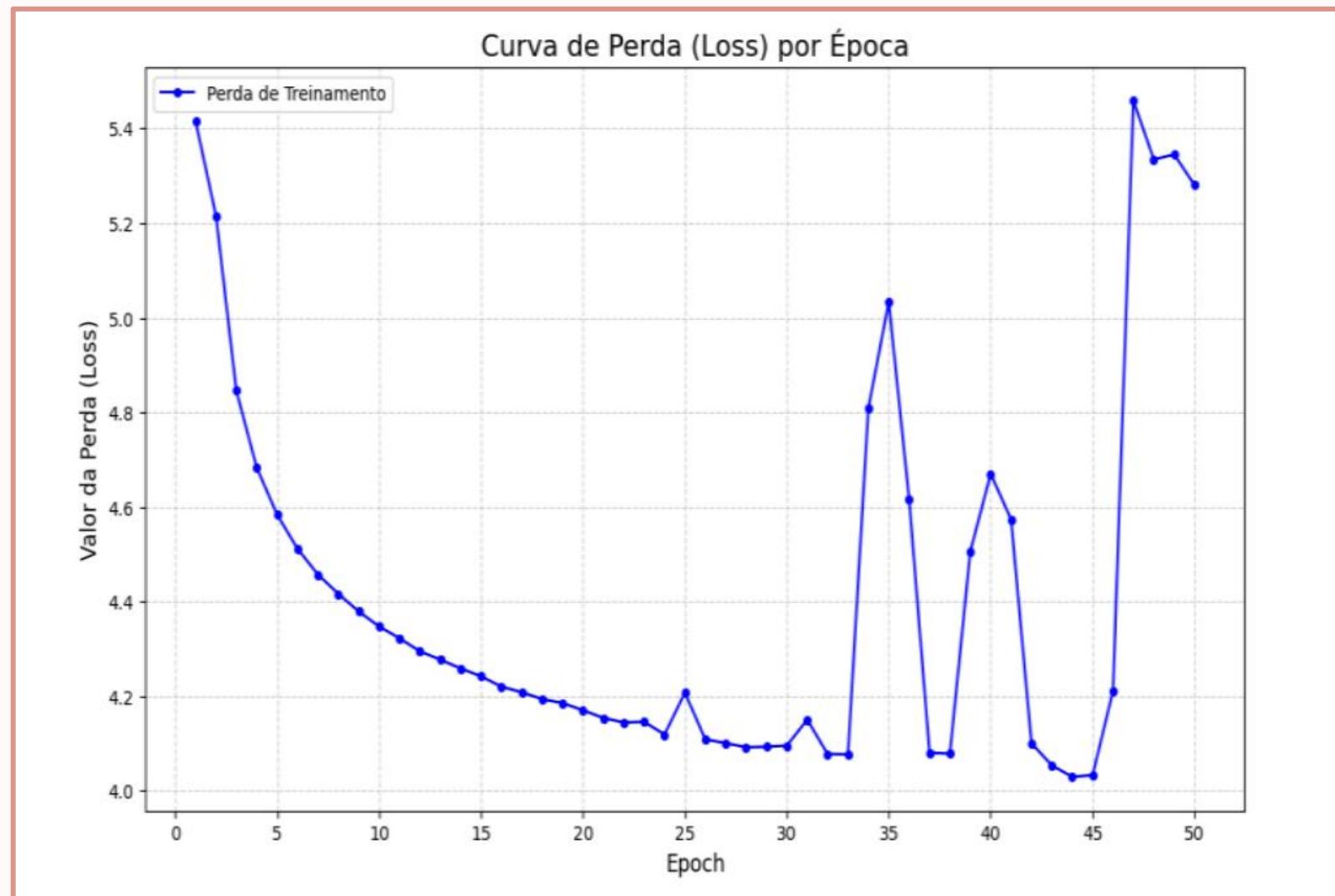
Modelo	Tempo de Execução	VRAM	RAM do Sistema
LSTM + Chopin	10min 50 seg	1.3 GB	2.5 GB
LSTM + MAESTRO	4h 41min 53seg	17.3 GB	+100 GB
Transformer + Chopin	2min 38seg	7 GB	3 GB
Transformer + MAESTRO	1h 23min	7GB	5 GB

# Análise Quantitativa: Curvas de Perda



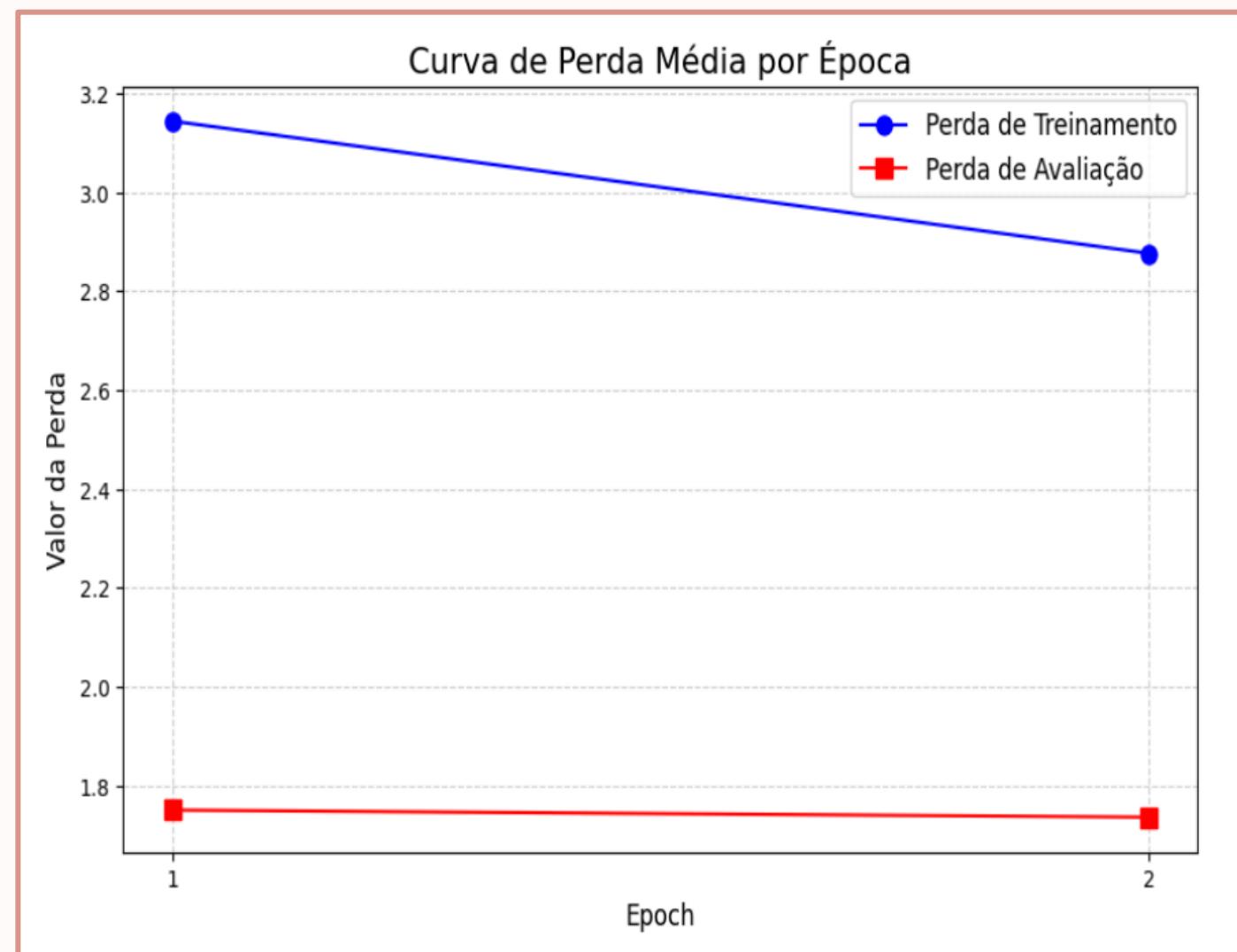
Curva de Perda por Epoch (LSTM com dataset Chopin)

# Análise Quantitativa: Curvas de Perda



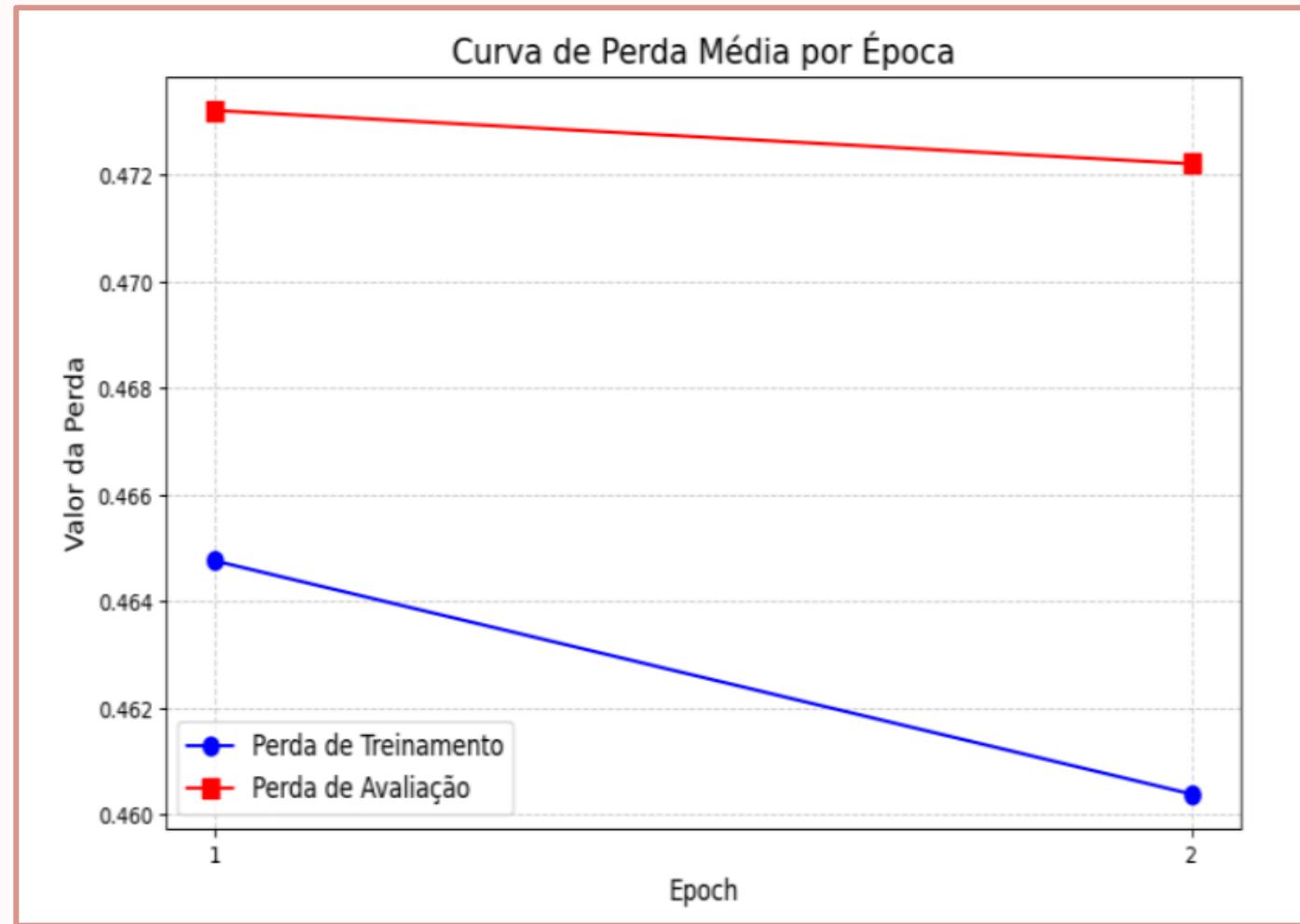
Curva de Perda por Epoch (LSTM com dataset Maestro)

# Análise Quantitativa: Curvas de Perda



Curva de Perda e Avaliação por Epoch (Transformers com dataset Chopin)

# Análise Quantitativa: Curvas de Perda

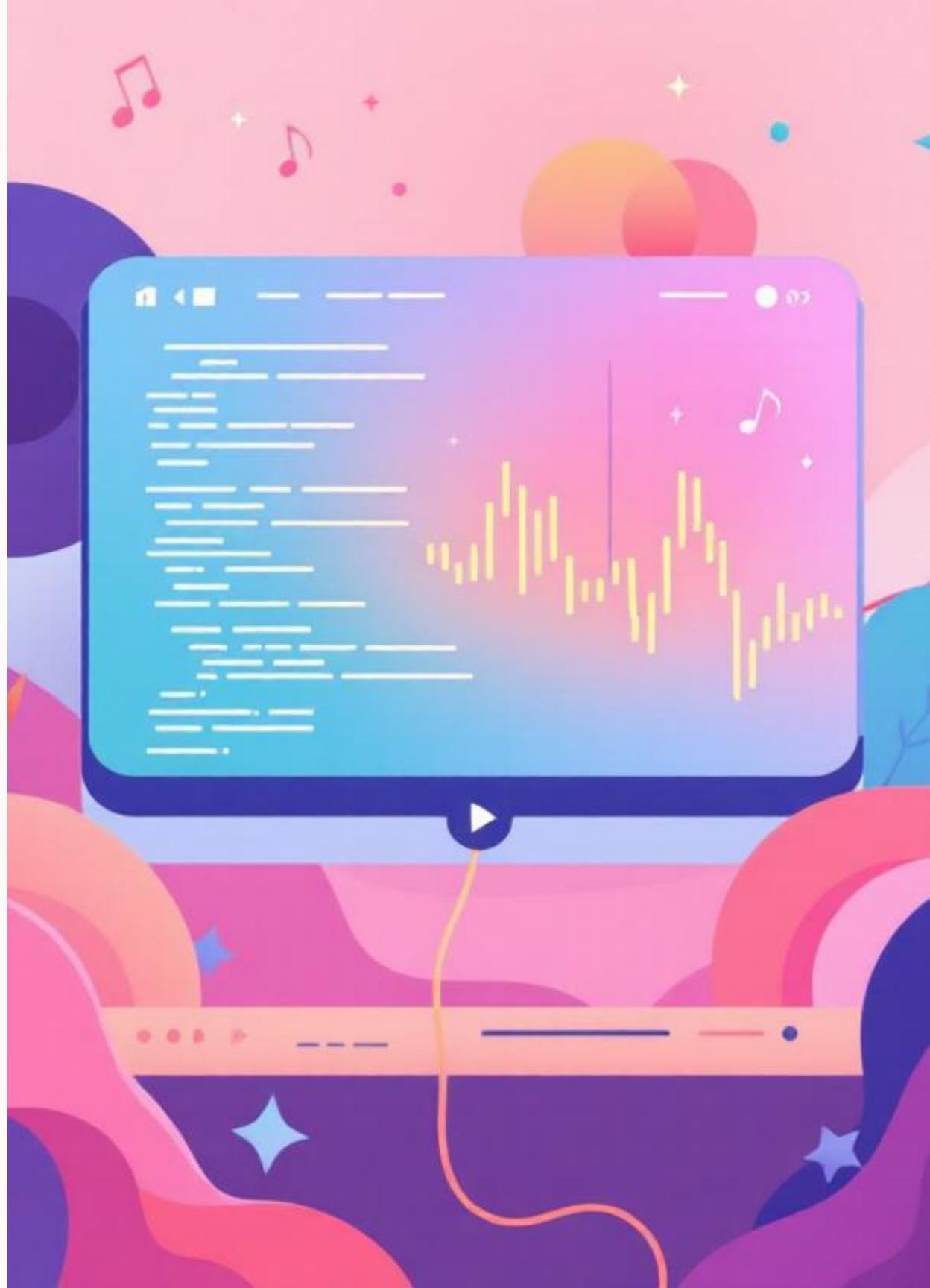


Curva de Perda e Avaliação por Epoch (Transformers com dataset Maestro)

# Desafios de Implementação

## *Gargalo LSTM: Pré-processamento*

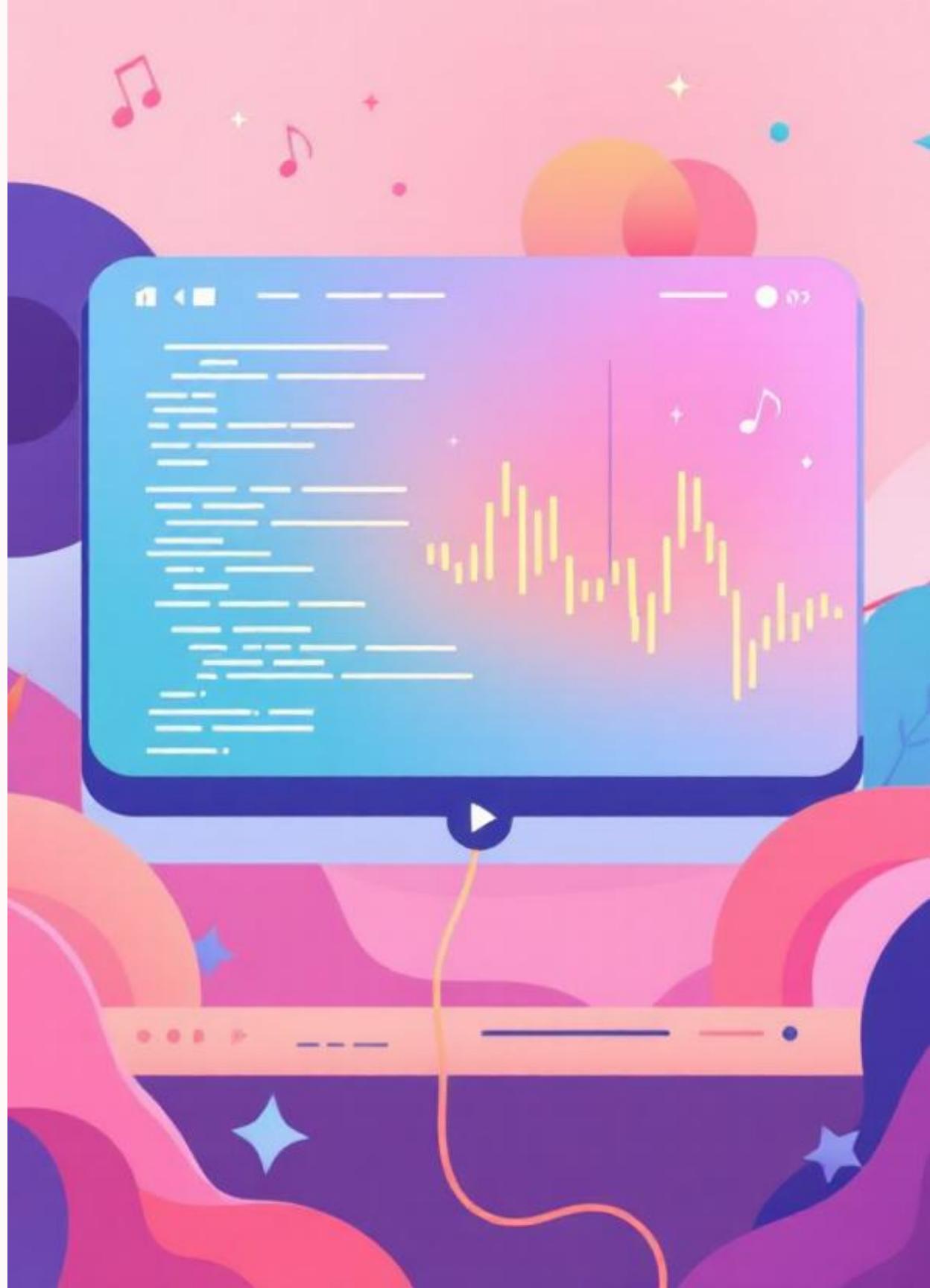
- Pré-processamento com a biblioteca music21
- Maestro: pico de > 100Gb de RAM
- Inviabiliza ambientes de baixo custo



# Desafios de Implementação

## *Gargalo Transformers: Treinamento*

- RAM da GPU (VRAM) de aproximadamente 7 Gb
- Recurso caro
- Impacta eficiência do projeto

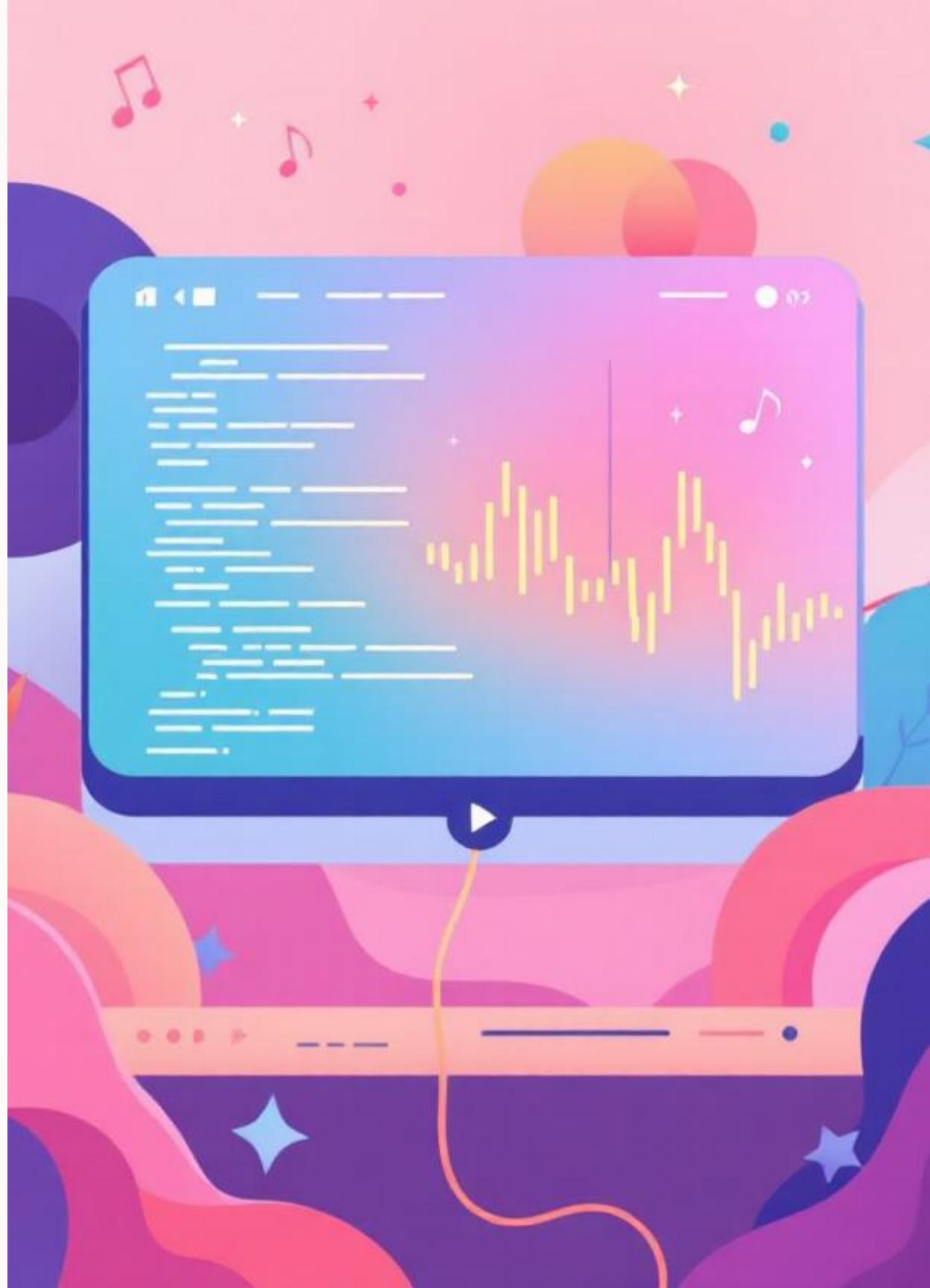


# Desafios de Implementação



## *Solução: Colab Pro*

- Migração essencial para garantir viabilidade
- Permite alocar recursos computacionais avançados



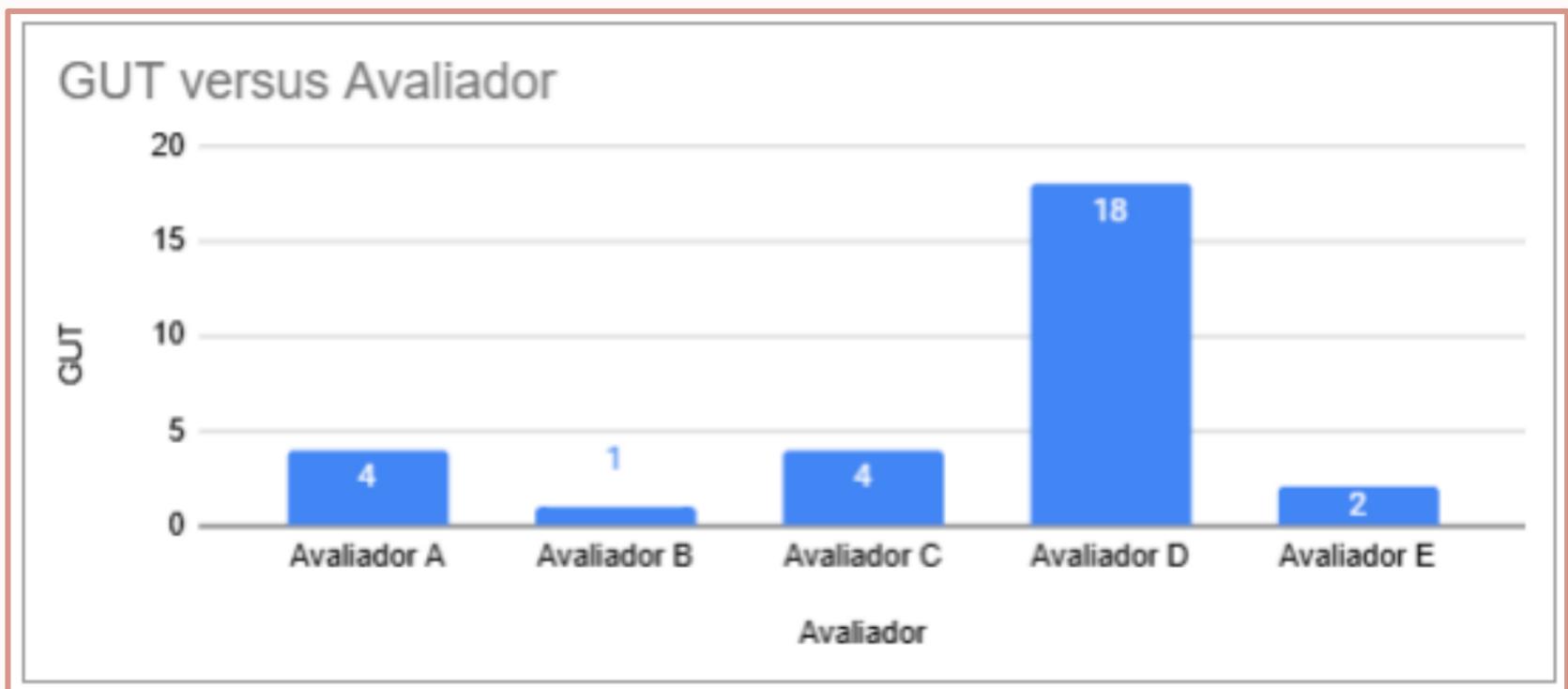
# Análise Qualitativa: Resultados da Linha de Base

## *LSTM (Chopin & MAESTRO)*

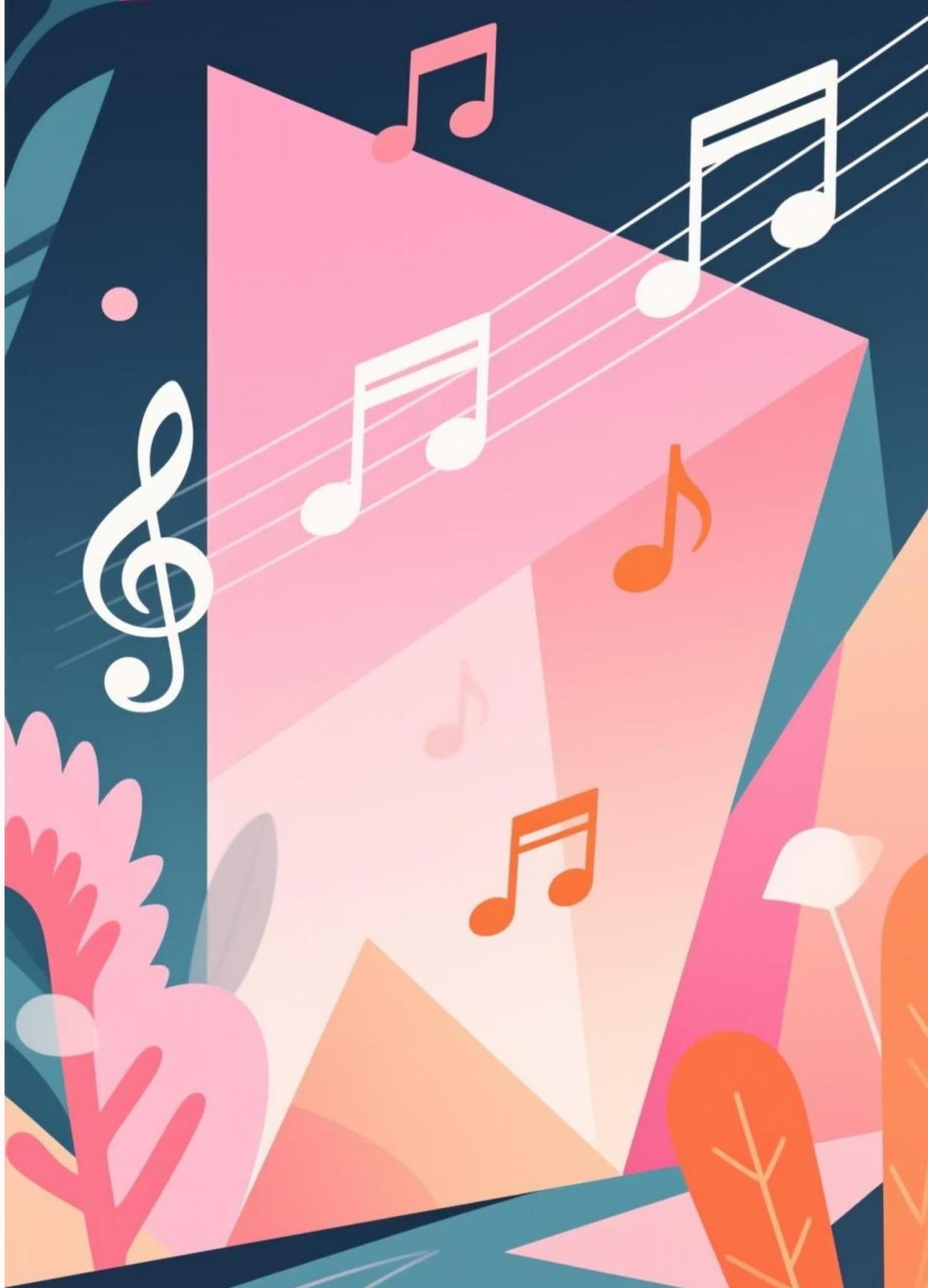
- Musicalmente pobre, atonal e rígido
- Falha ao convergir o MAESTRO de forma estável



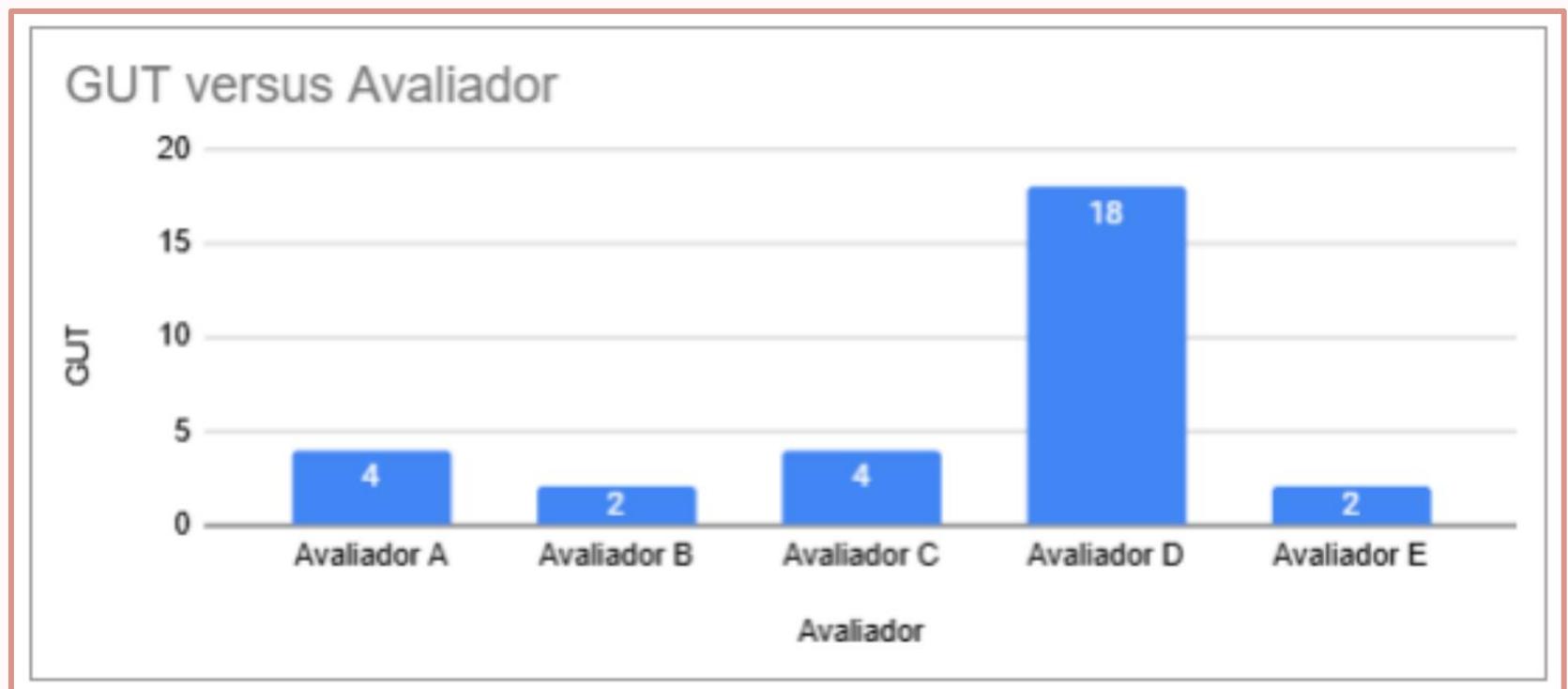
# Análise Qualitativa: Resultados da Linha de Base



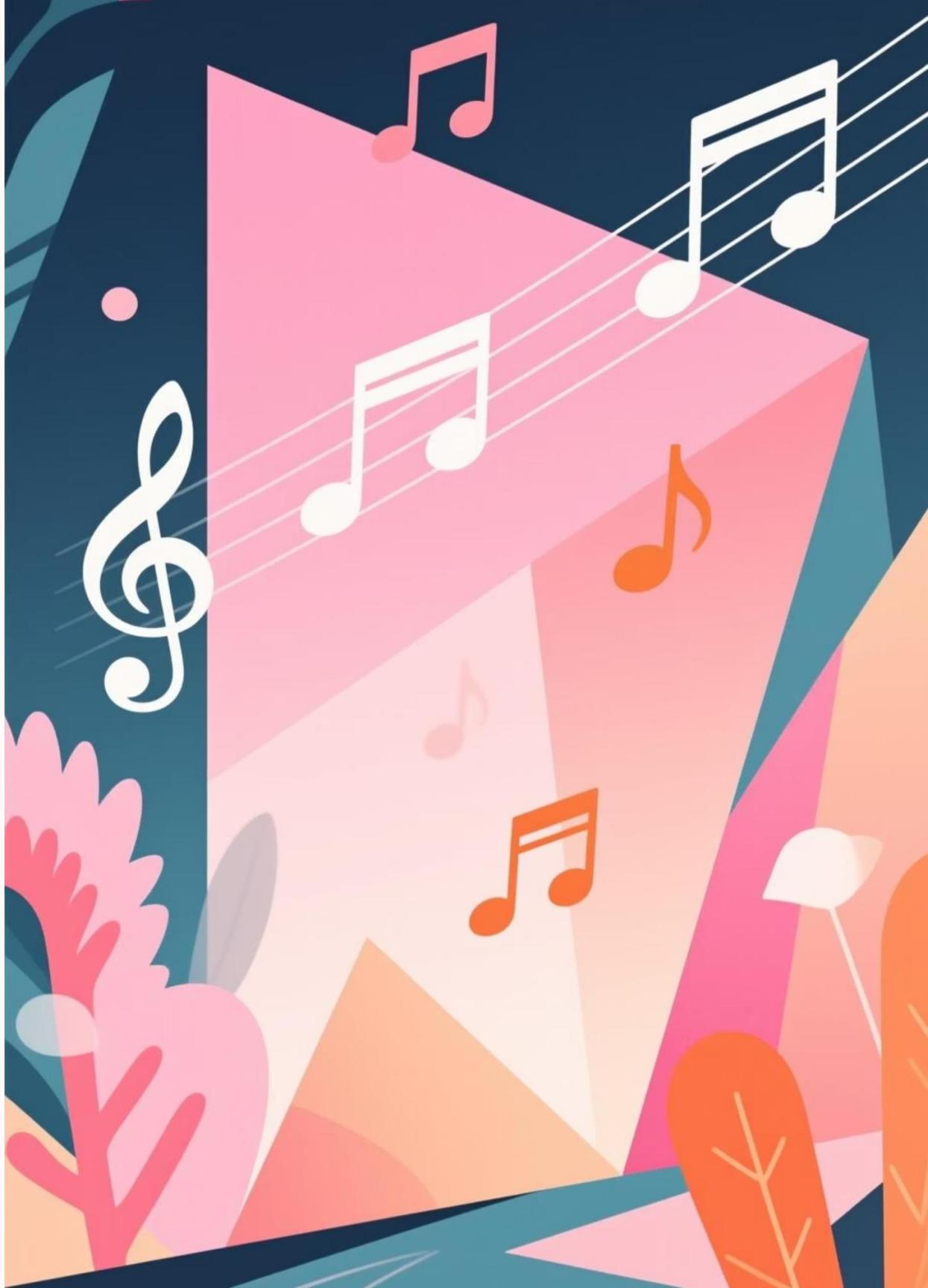
GUT vs Avaliador (LSTM + Chopin)



# Análise Qualitativa: Resultados da Linha de Base



GUT vs Avaliador (LSTM + Chopin



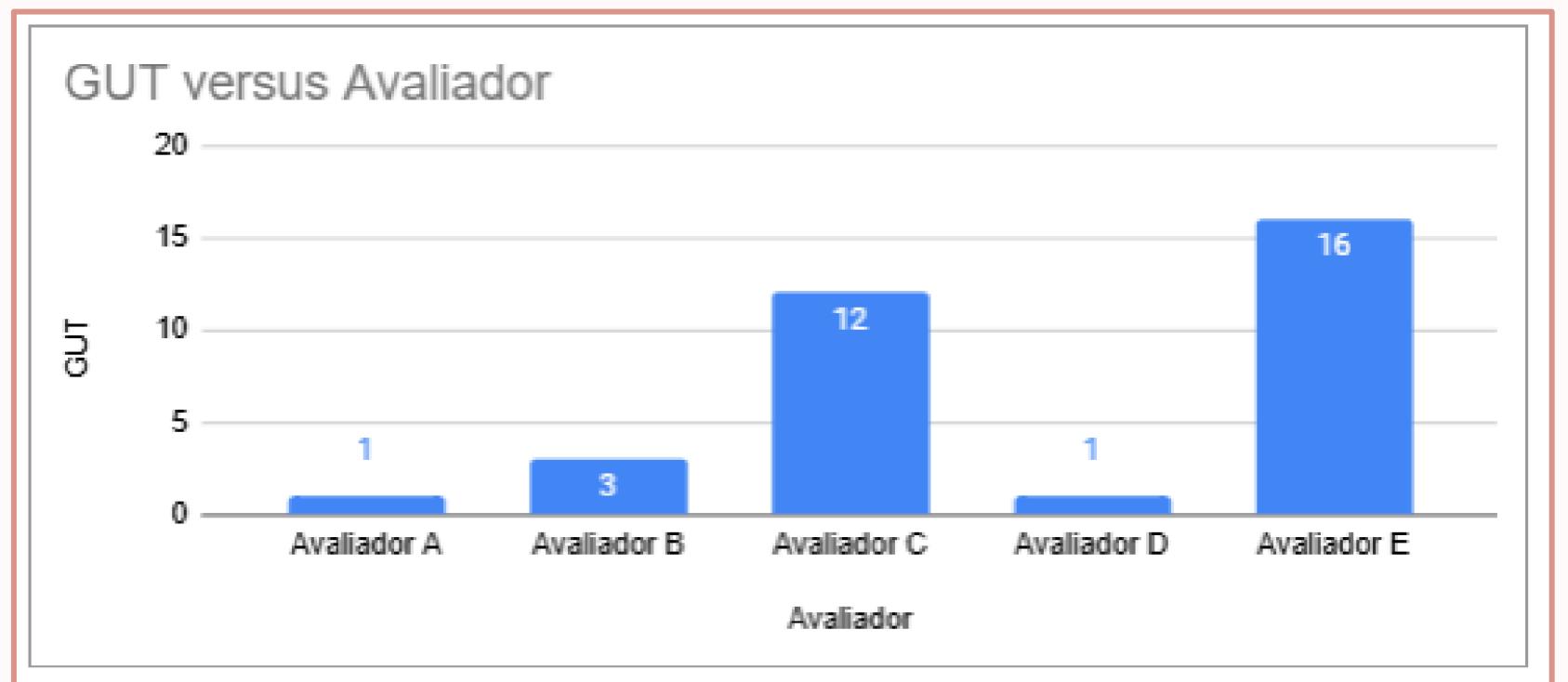
# Análise Qualitativa: Resultados da Linha de Base

## *Transformers + Chopin*

- Musical atonal, mas com complexidade rítmica
- Sintomas de underfitting



# Análise Qualitativa: Resultados da Linha de Base



GUT vs Avaliador (Transformers + Chopin)



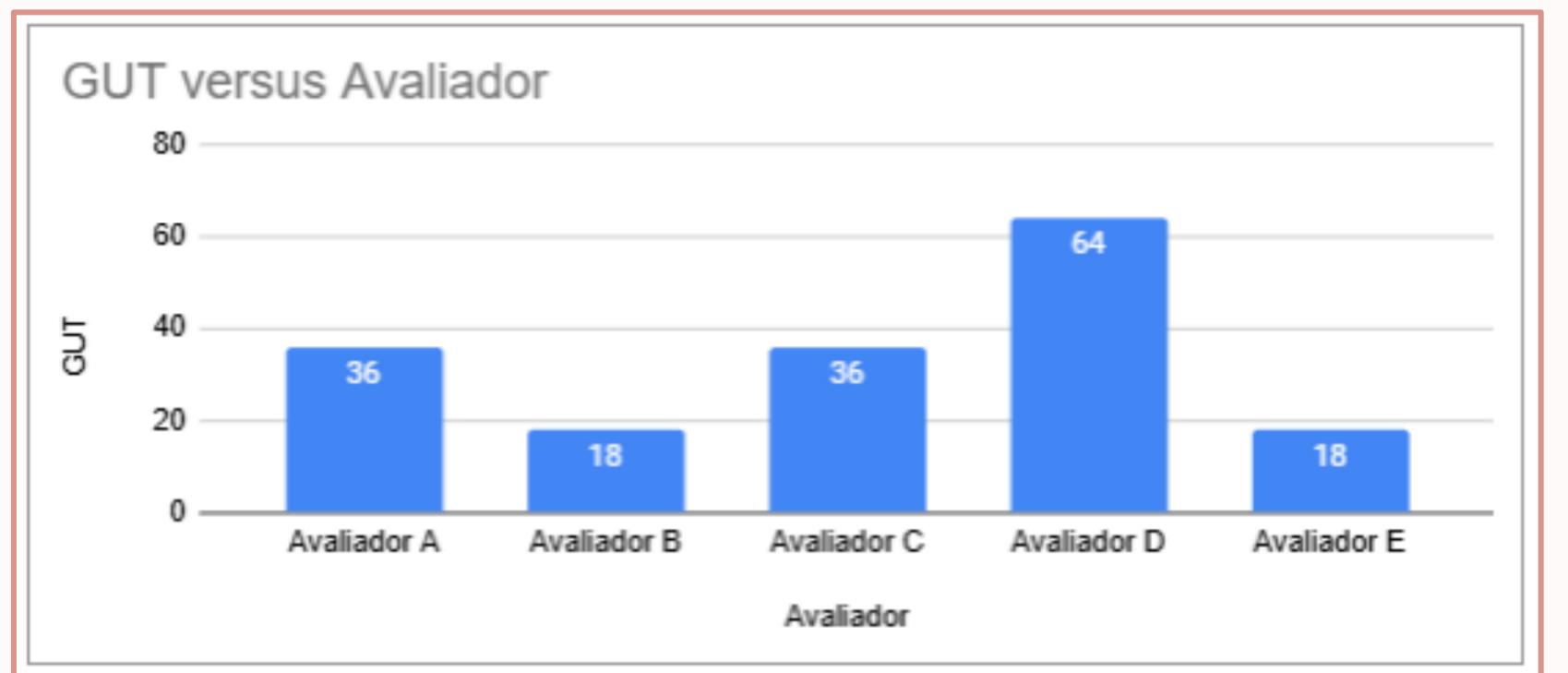
# Análise Qualitativa: Resultados da Linha de Base

## *Transformers + MAESTRO*

- Mais coerente e bem sucedido
- Peça tonal com melodia e harmonia clara



# Análise Qualitativa: Resultados da Linha de Base



GUT vs Avaliador (Transformers + Maestro)

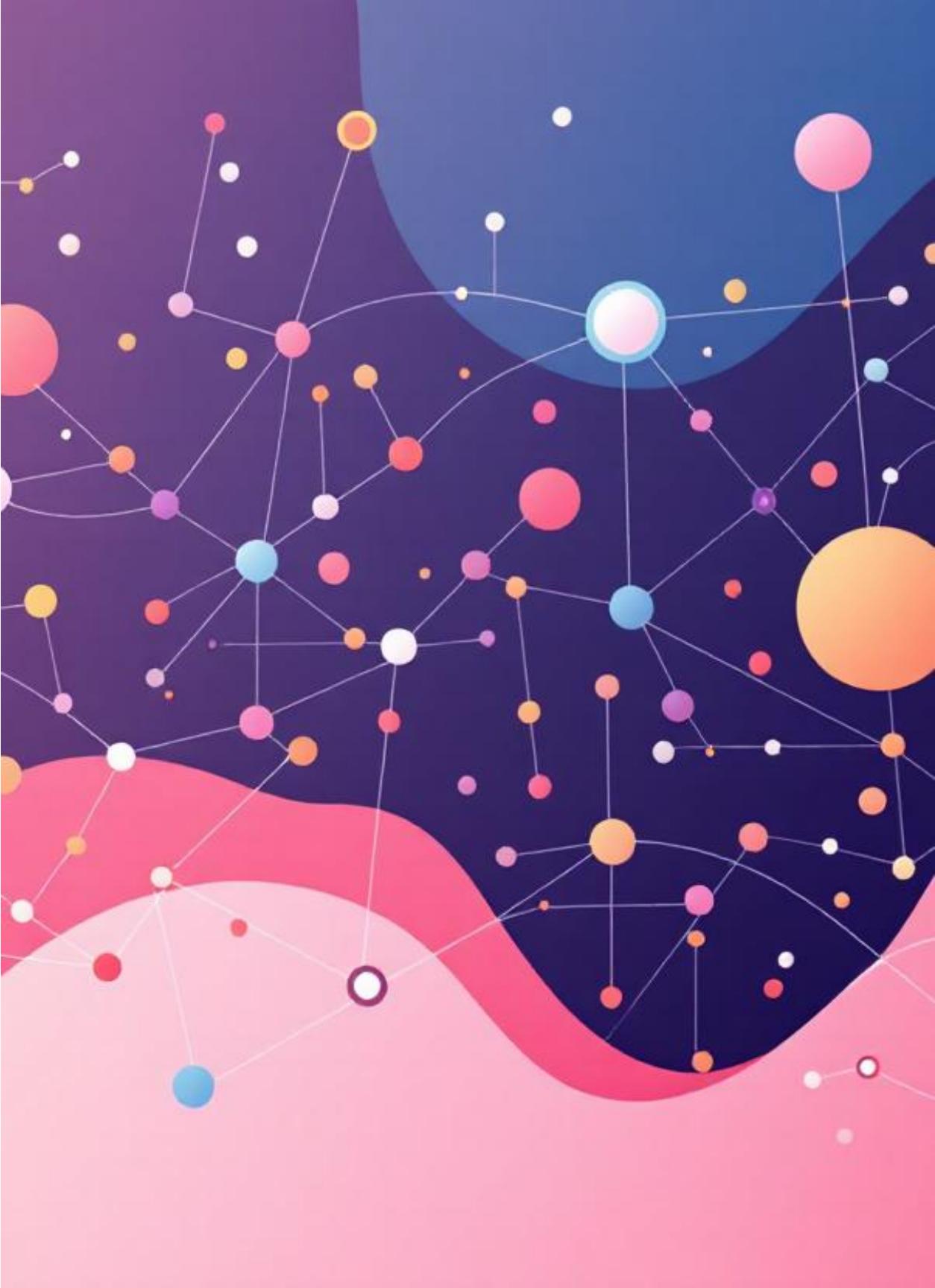


# Estratégias de Ajuste e Refinamento



## *Aumento da Profundidade*

- Adicionar camadas LSTM
- Aumentar largura



# Estratégias de Ajuste e Refinamento



## *Extensão do Treinamento*

- Aumentar número de Epochs



# Estratégias de Ajuste e Refinamento



## *Otimizador e Taxa de Aprendizado*

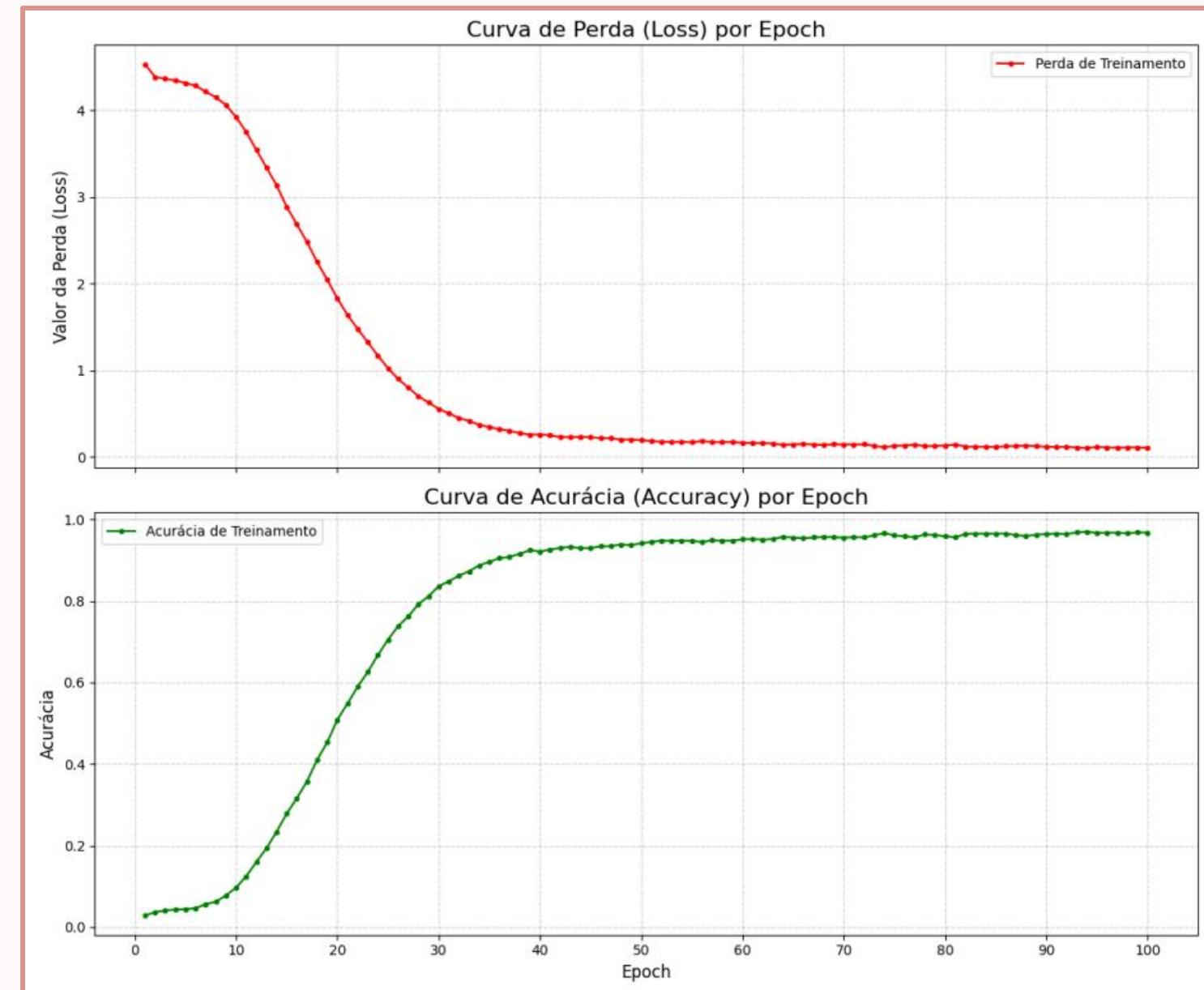
- Substituir Adamax por Adam/AdamW
- Taxa de aprendizado para estabilizar treinamento



# Rodada 2: Custo Computacional dos Ajustes

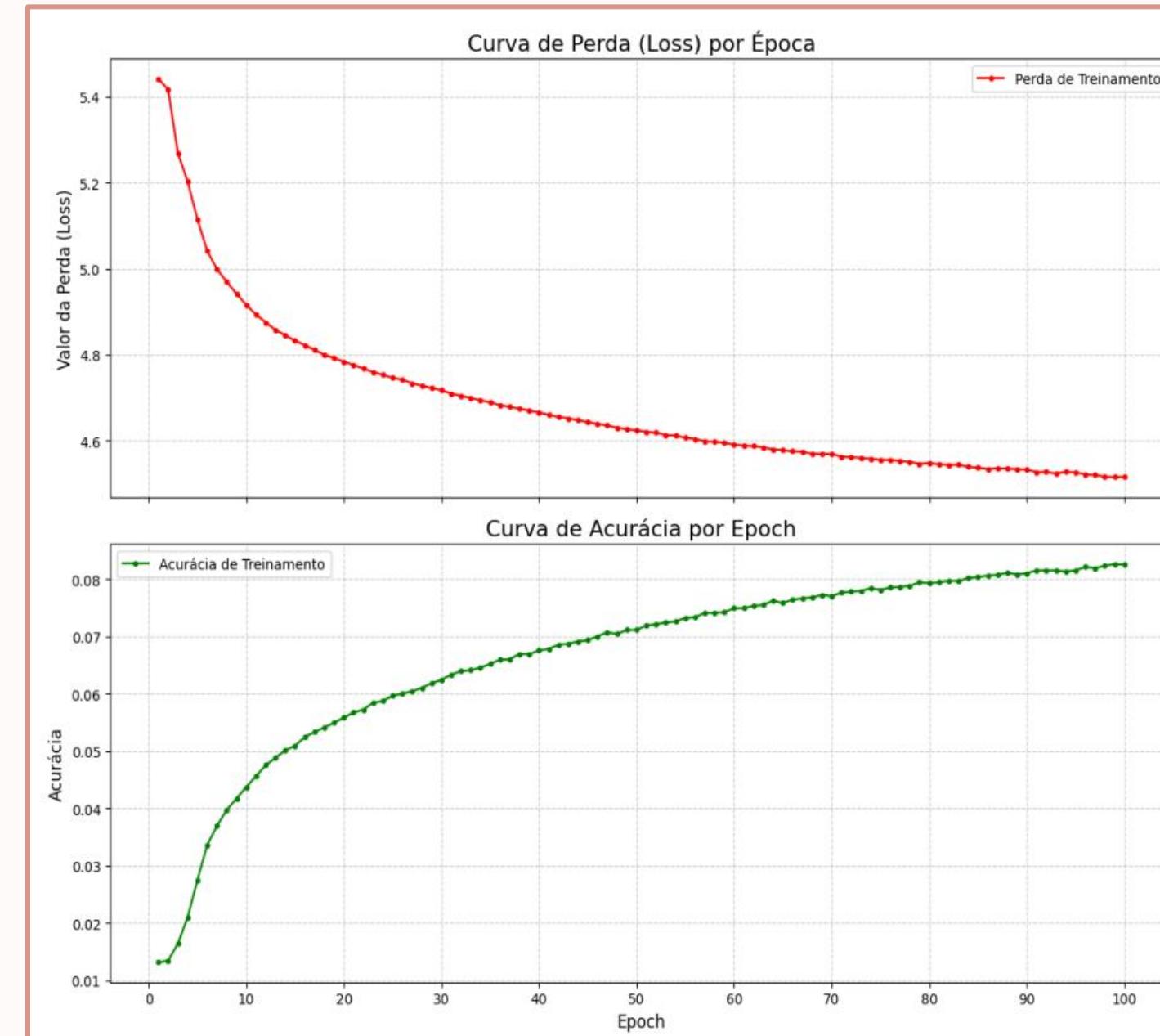
Modelo	Tempo de Execução (v2)	VRAM	RAM do Sistema
LSTM + Chopin (v2)	52min 35seg	2.4 GB	2.8 GB
LSTM + MAESTRO (v2)	6h 51min 57seg	17.5 GB	80 GB
Transformer + Chopin (v2)	15min 11seg	7 GB	2.5 GB
Transformer + MAESTRO (v2)	5h 8min 49seg	9 GB	4.5 GB

# Análise Quantitativa: Curvas de Perda/Acurácia



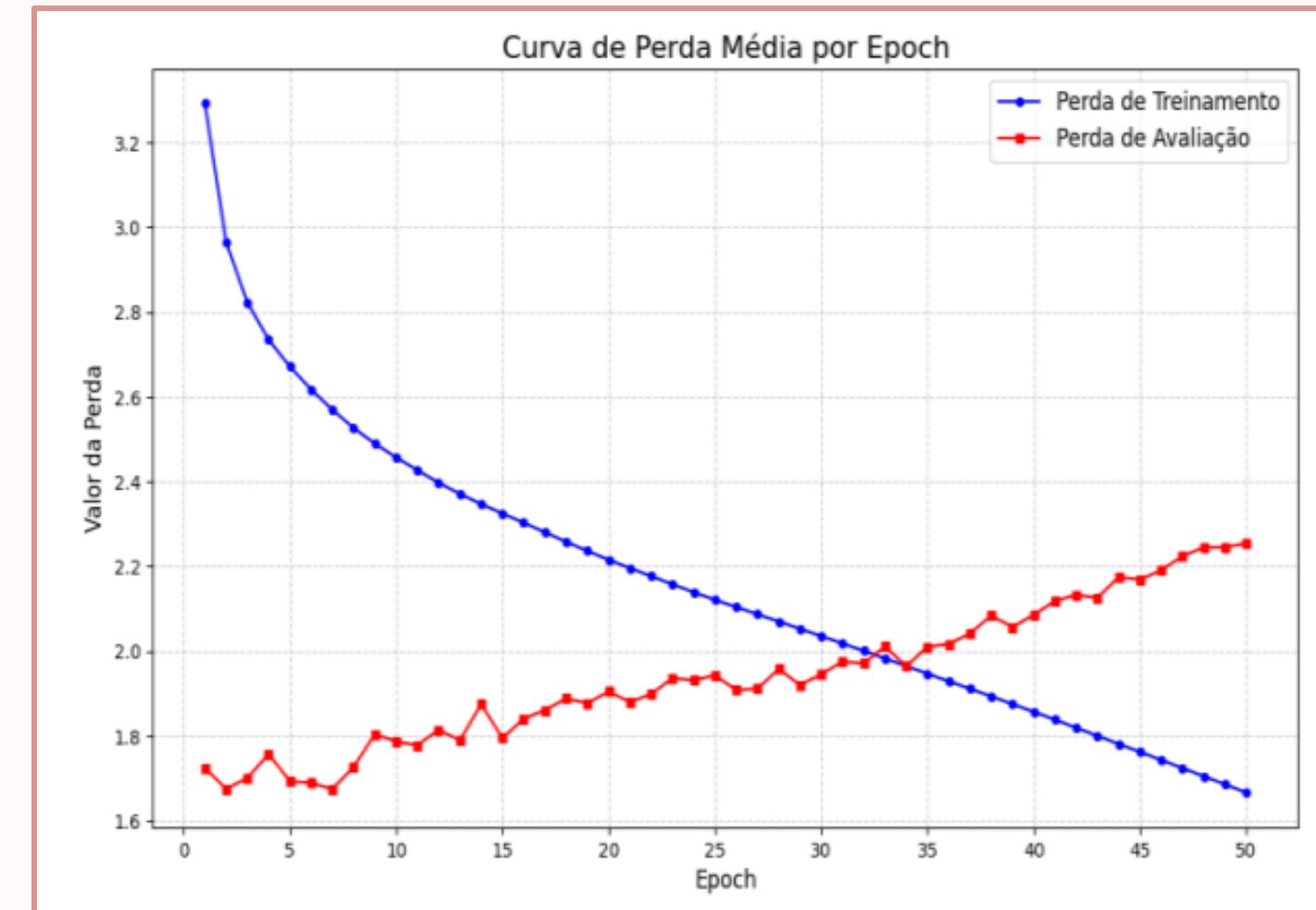
Curva de Perda e Acurácia (LSTM com ajustes + dataset Chopin)

# Análise Quantitativa: Curvas de Perda/Acurácia



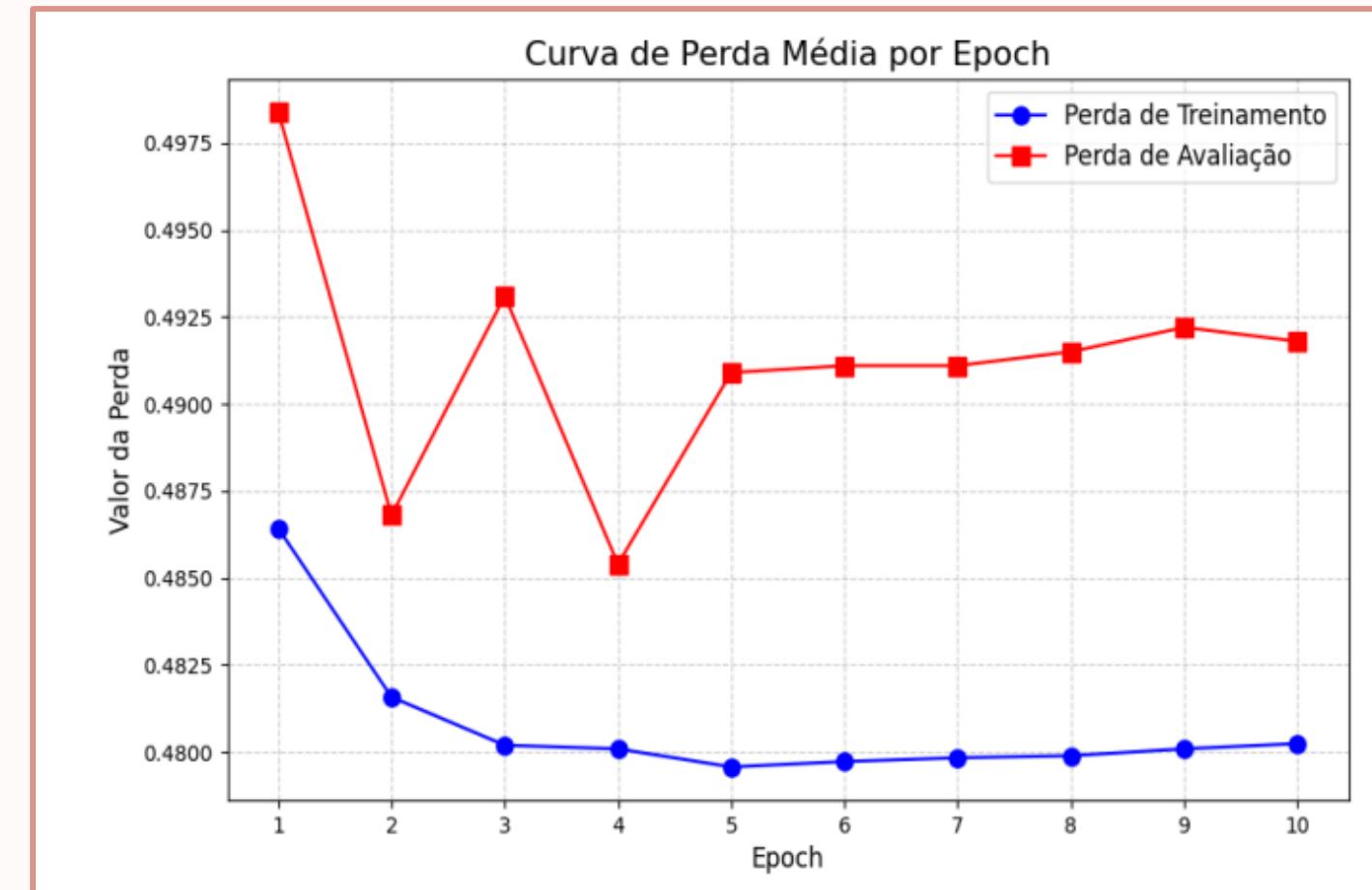
Curva de Perda e Acurácia (LSTM com ajustes + dataset Maestro)

# Análise Quantitativa: Curvas de Perda/Acurácia



Curva de Perda e Avaliação (Transformers com ajustes + dataset Chopin)

# Análise Quantitativa: Curvas de Perda/Acurácia



Curva de Perda e Avaliação (Transformers com ajustes + dataset Maestro)

# Rodada 2: O Colapso do LSTM

## *LSTM + Chopin (v2)*

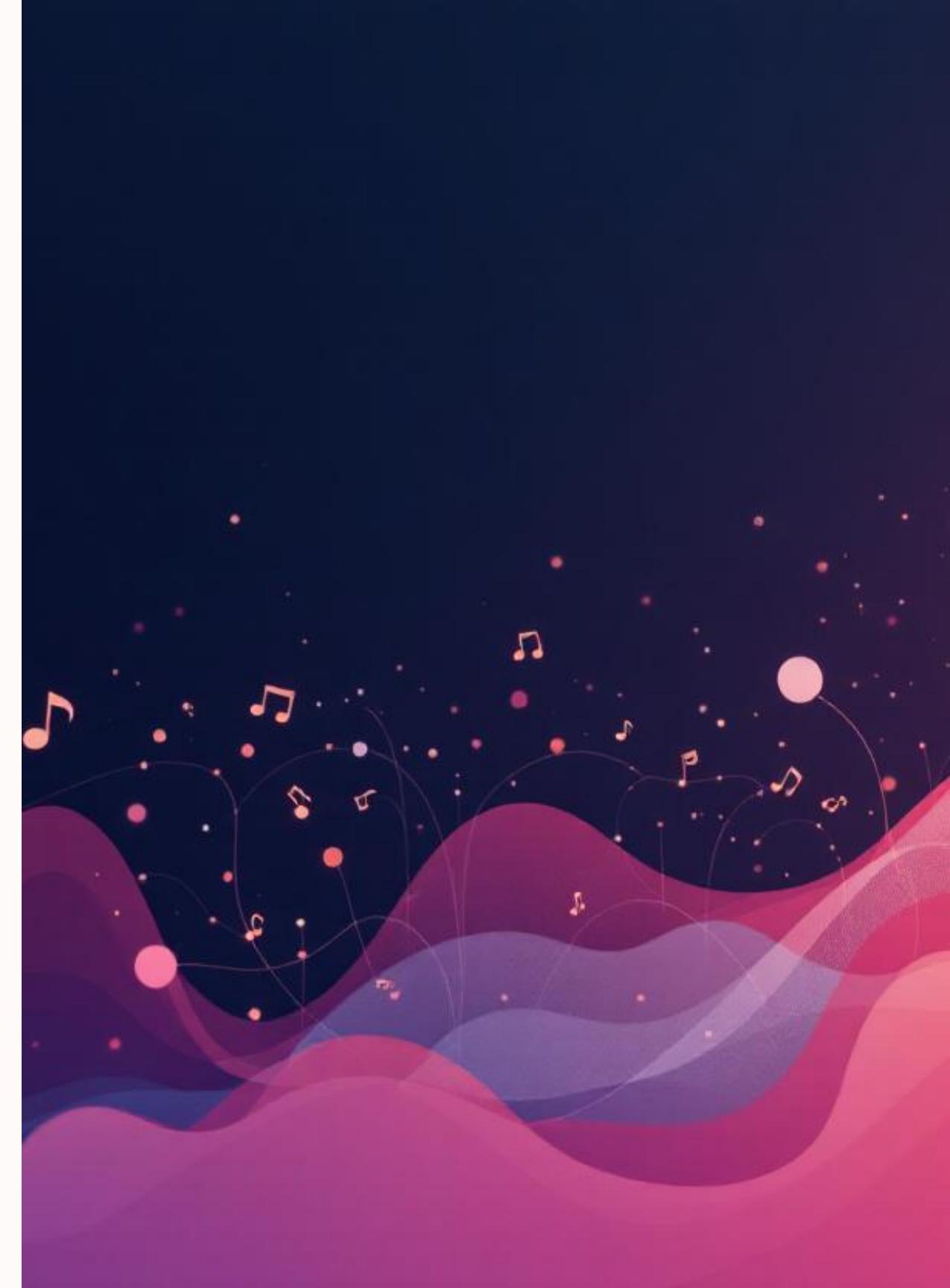
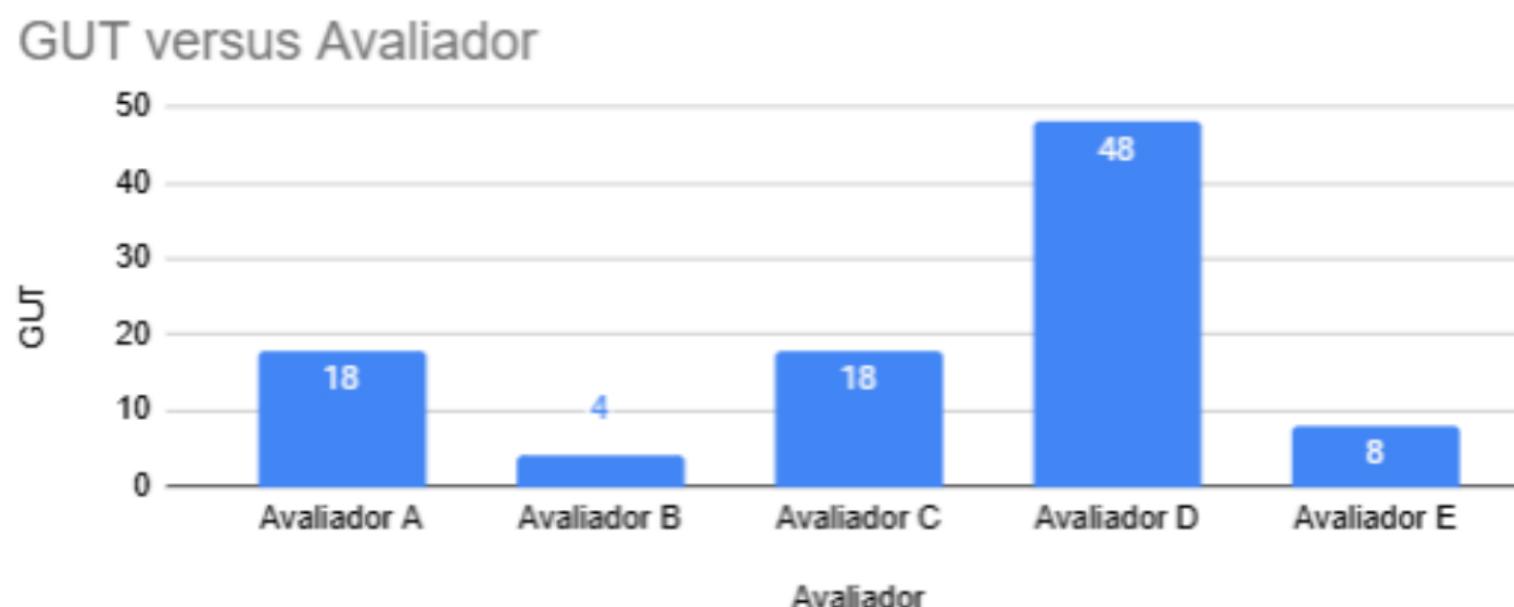
Leve melhora. Mais tonal, mas muito repetitivo.

Métrica GUT: 17.6



# Rodada 2: O Colapso do LSTM

GUT vs Avaliador (LSTM com ajustes + dataset Chopin)



# Rodada 2: O Colapso do LSTM

*LSTM + MAESTRO (v2)*

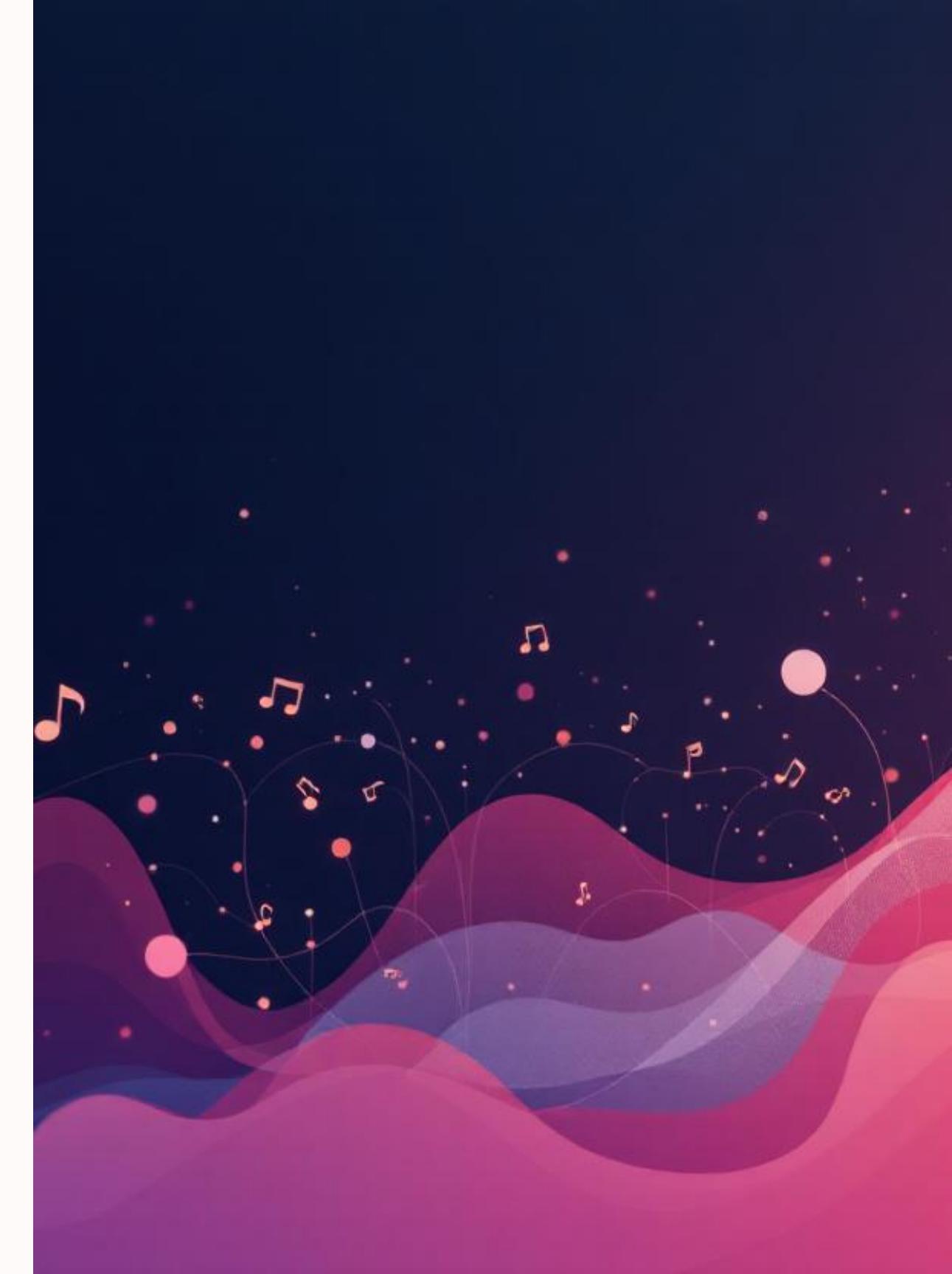
Falha total (Colapso de Modelo).

Nota repetida, indicando estagnação e ineficácia



# Rodada 2: O Colapso do LSTM

GUT vs Avaliador (LSTM com ajustes + dataset Maestro)



# Rodada 2: Convergência do Transformer

## Transformer + Chopin (v2)

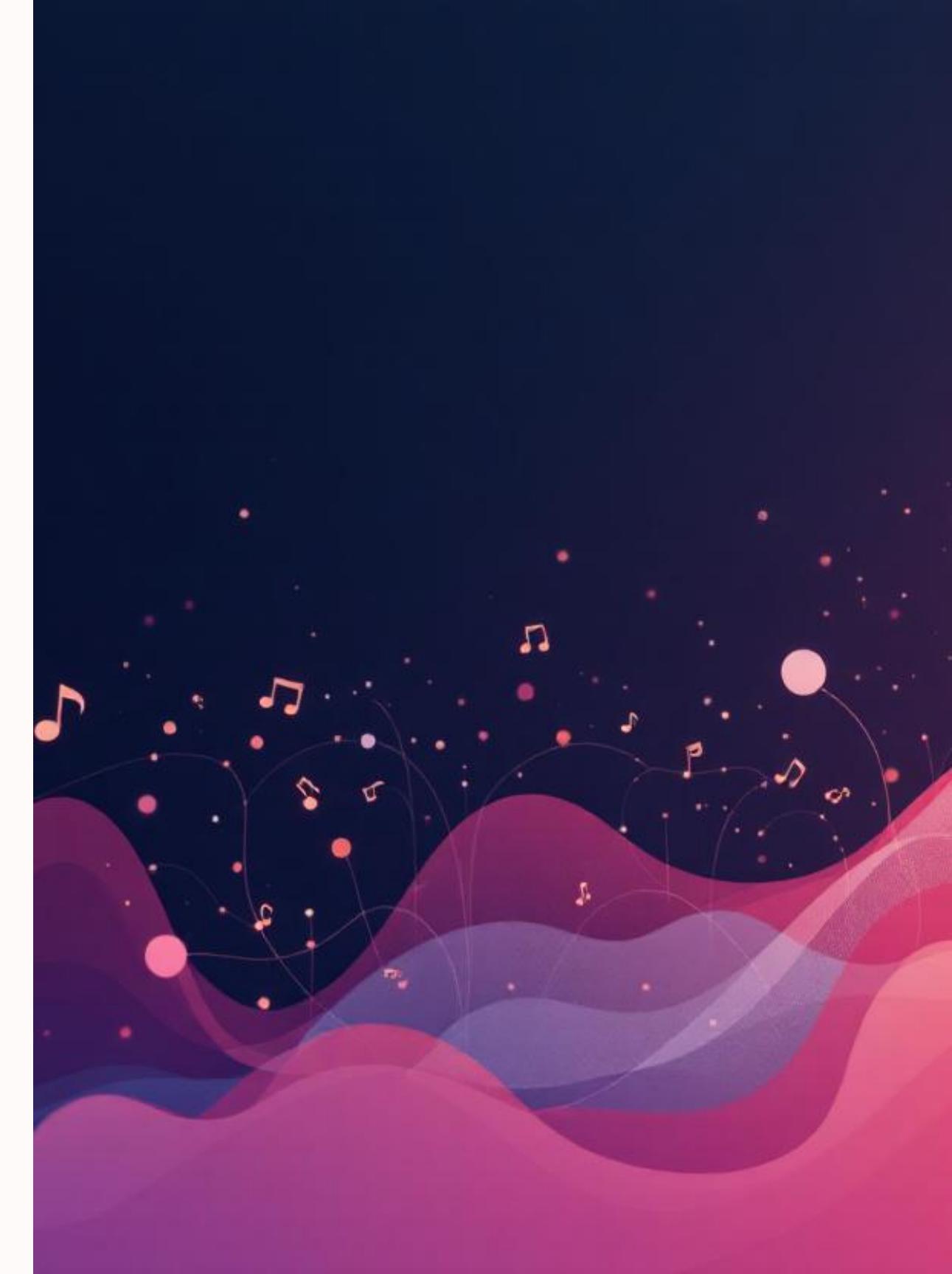
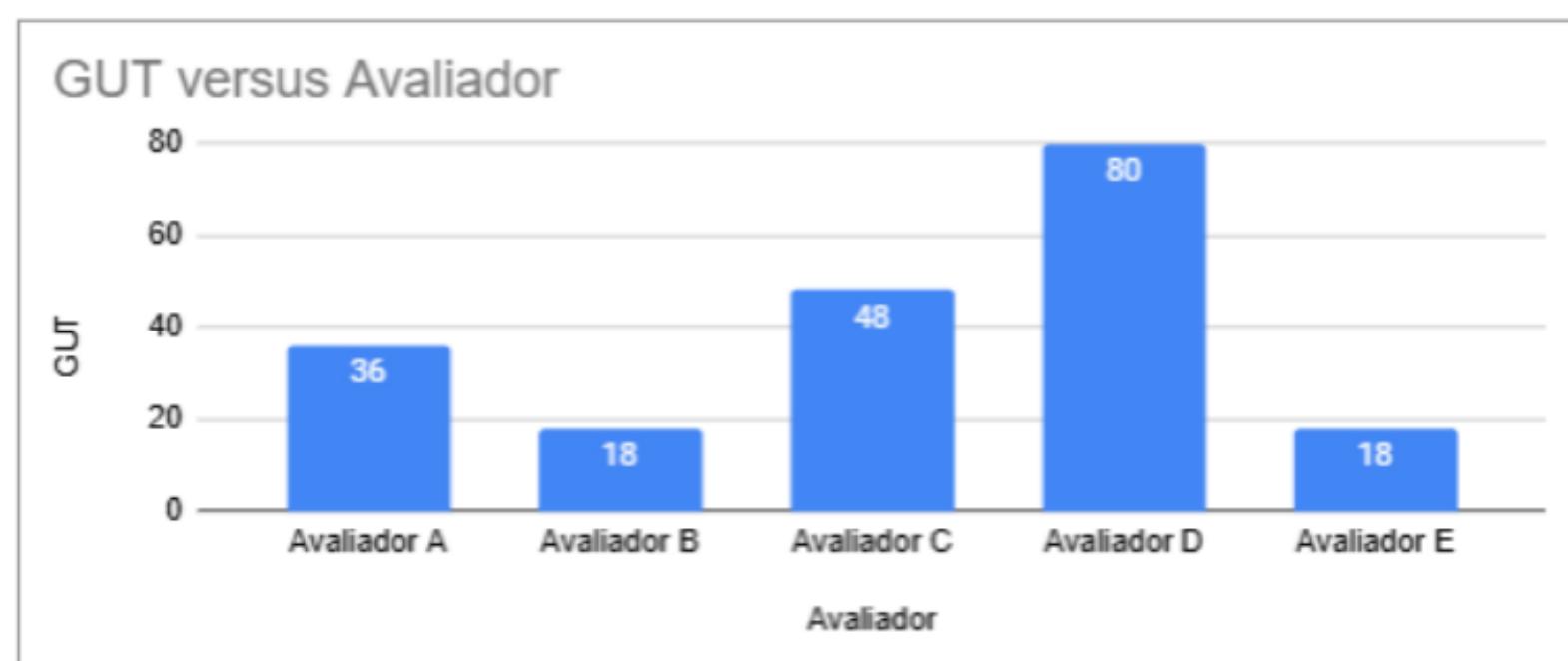
Convergiu para uma peça “minimalista” e coesa  
Representa evolução

Métrica GUT: 38.0



# Rodada 2: Convergência do Transformer

GUT vs Avaliador (Transformer com ajustes + dataset Chopin)



# Rodada 2: Convergência do Transformer

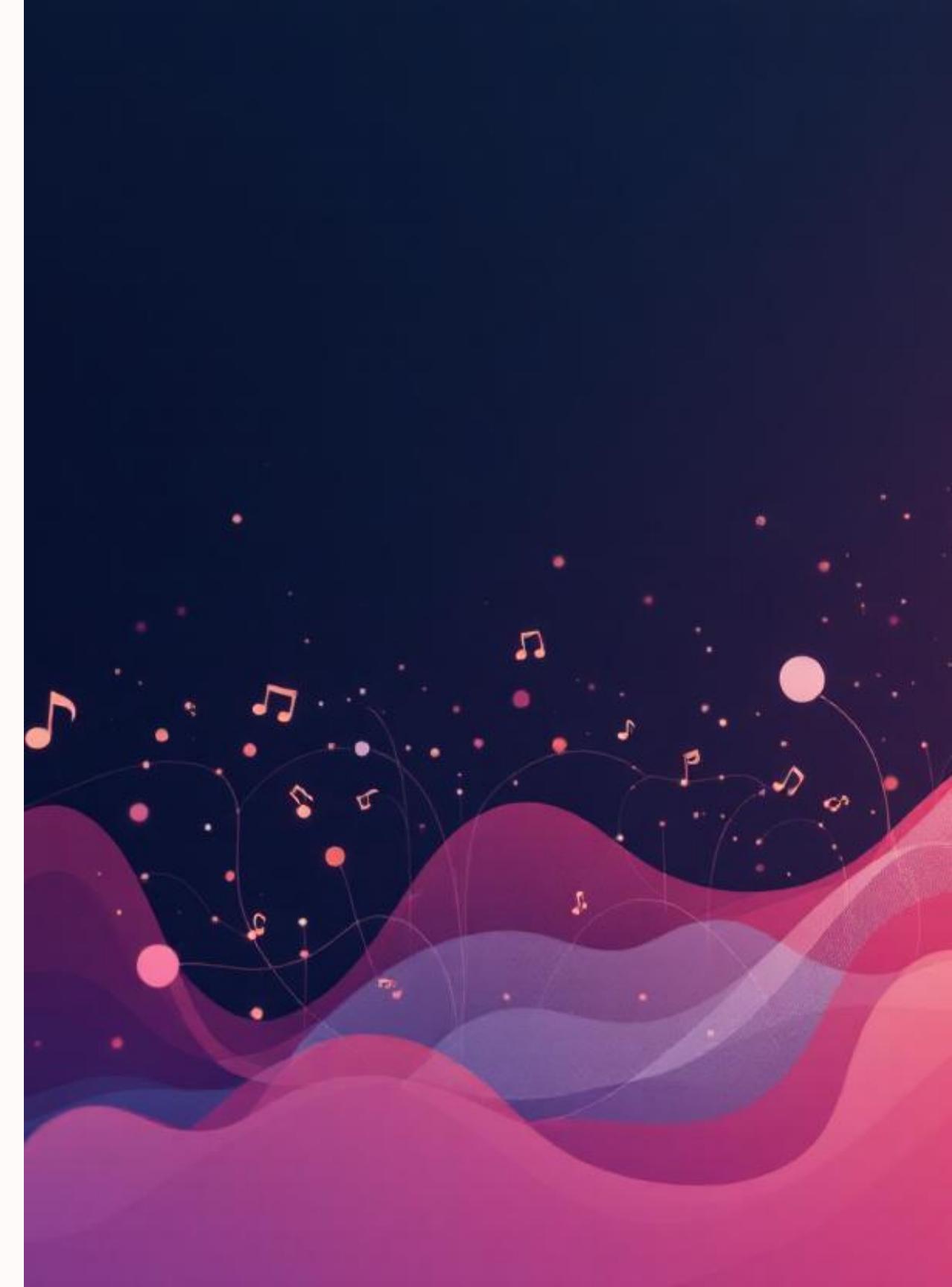
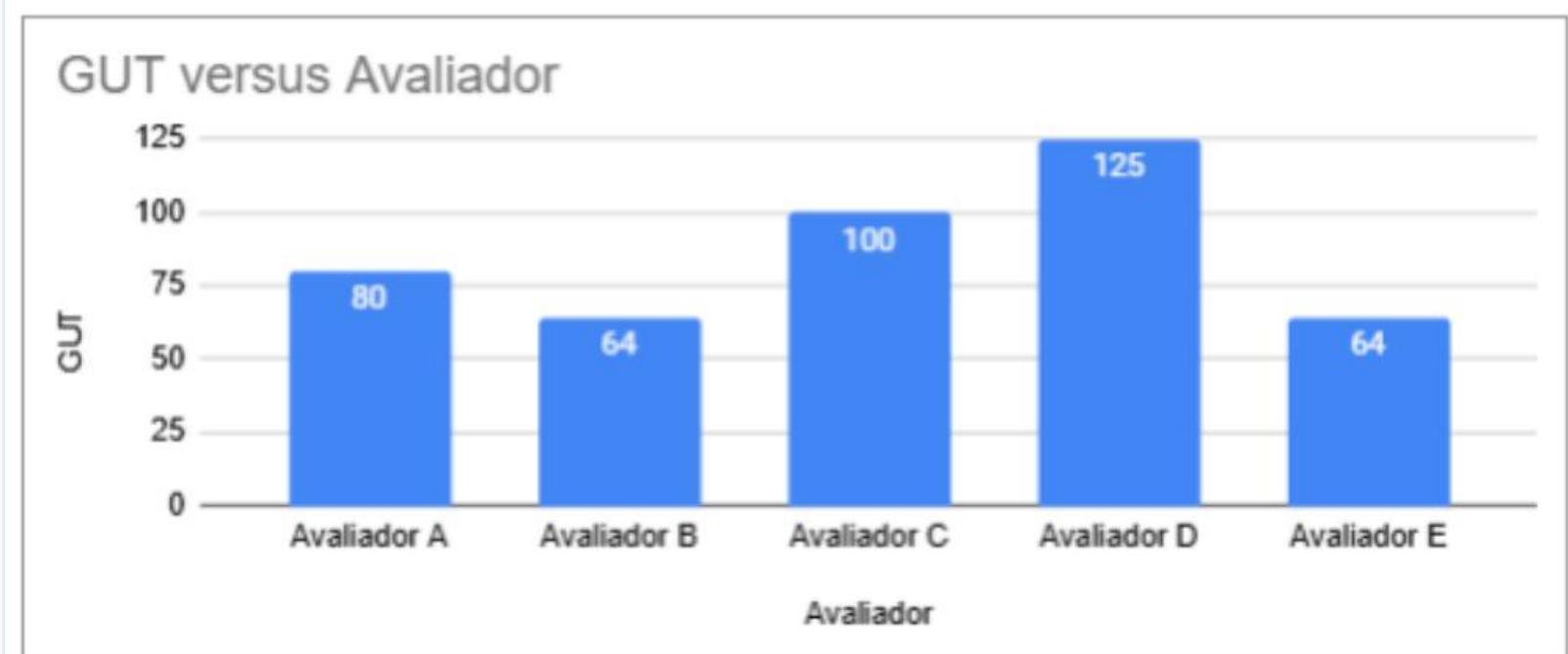
Transformer + MAESTRO (v2)

Melhor resultado do estudo  
Musicalmente completa e coerente  
· Métrica GUT: 86.6

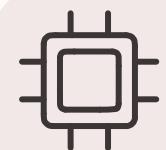


# Rodada 2: Convergência do Transformer

GUT vs Avaliador (Transformer com ajustes + dataset Maestro)

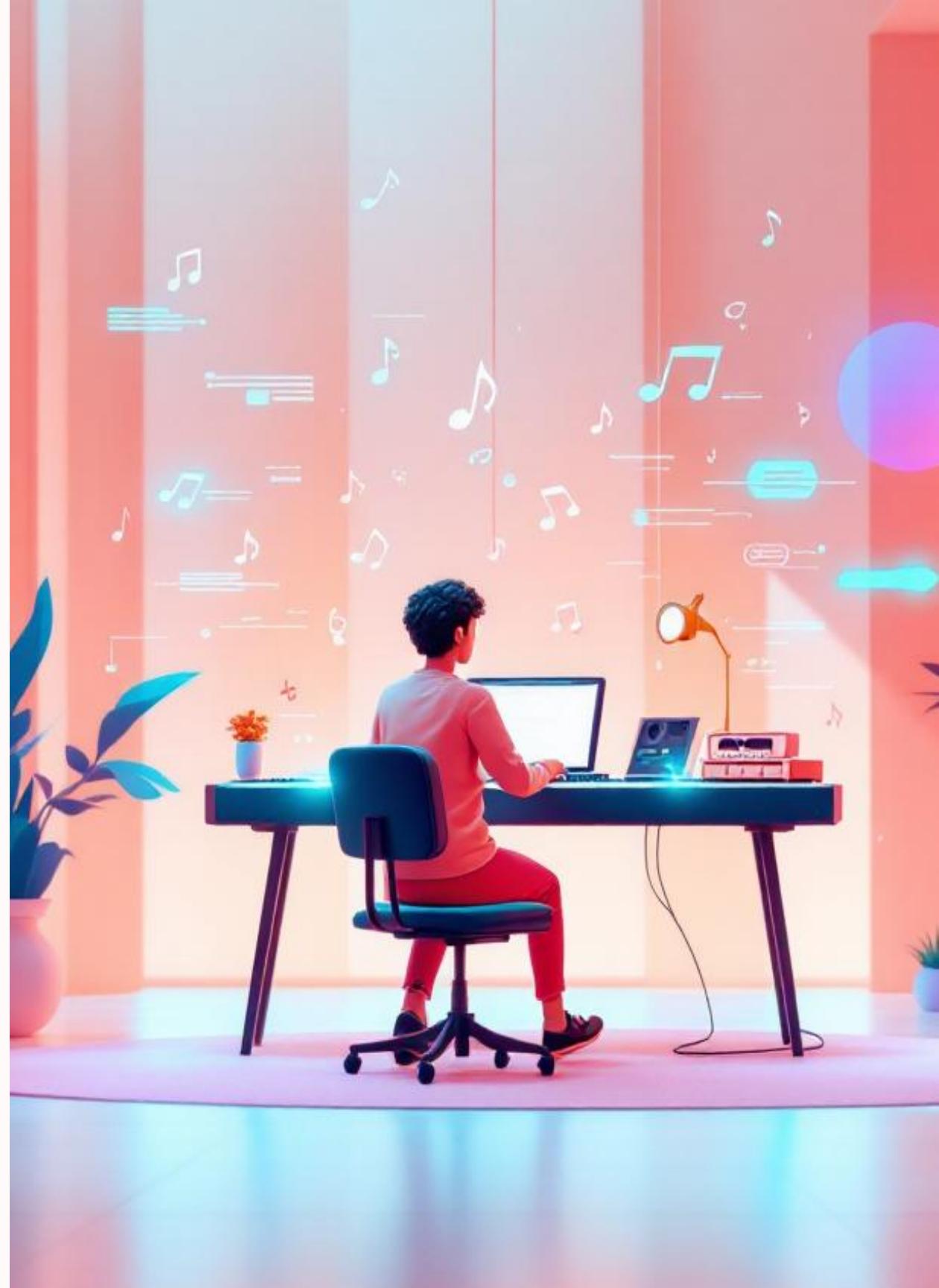


# Conclusões



## *Eficiência Superior*

- Transformer demonstrou ser a arquitetura mais eficaz para a tarefa de geração musical, superando o LSTM em complexidade e qualidade.



# Conclusões



## *Dados Essenciais*

- Apenas arquitetura não garante bom funcionamento;
- Riqueza do dataset é crucial para qualidade do resultado.

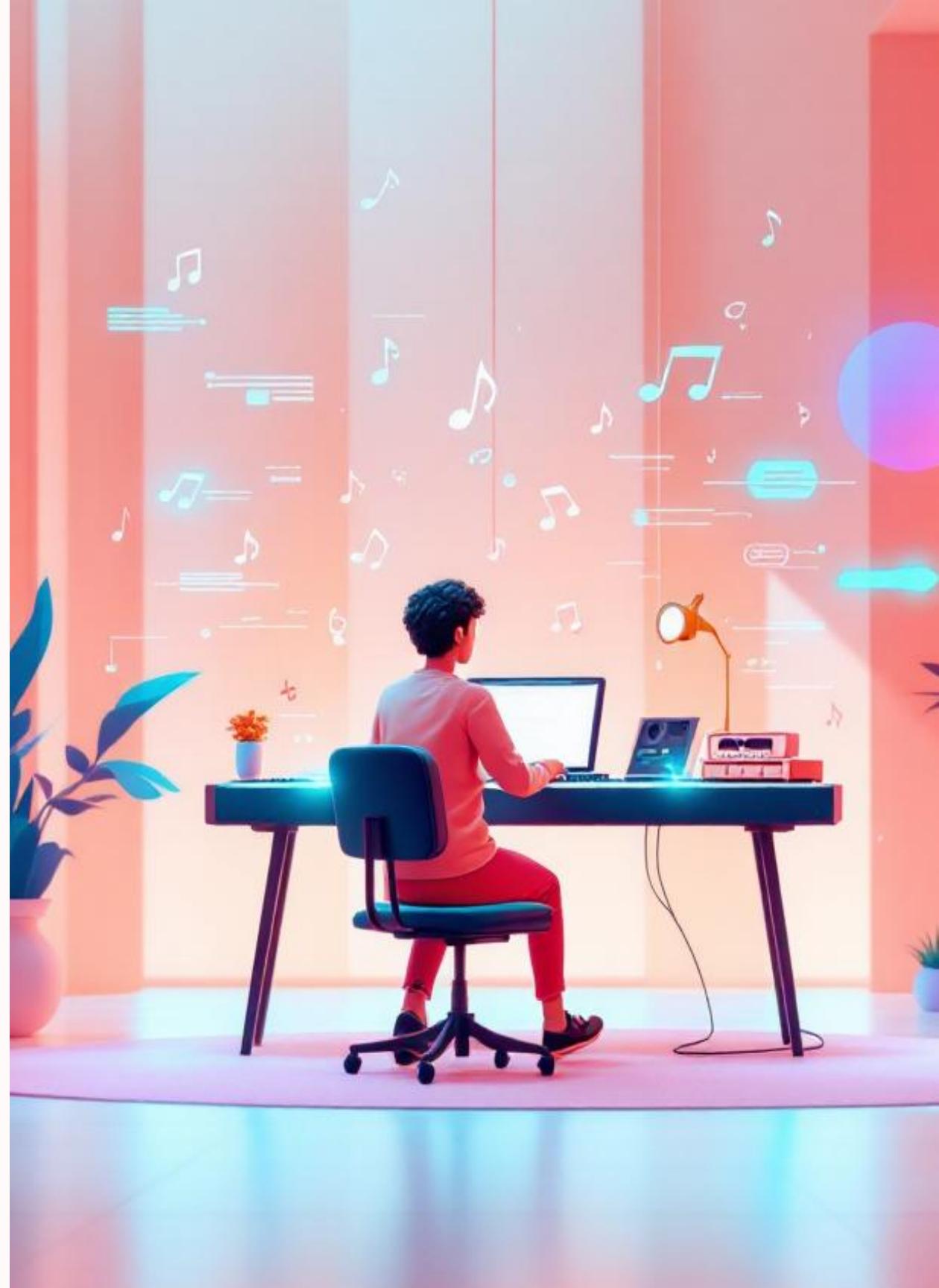


# Conclusões



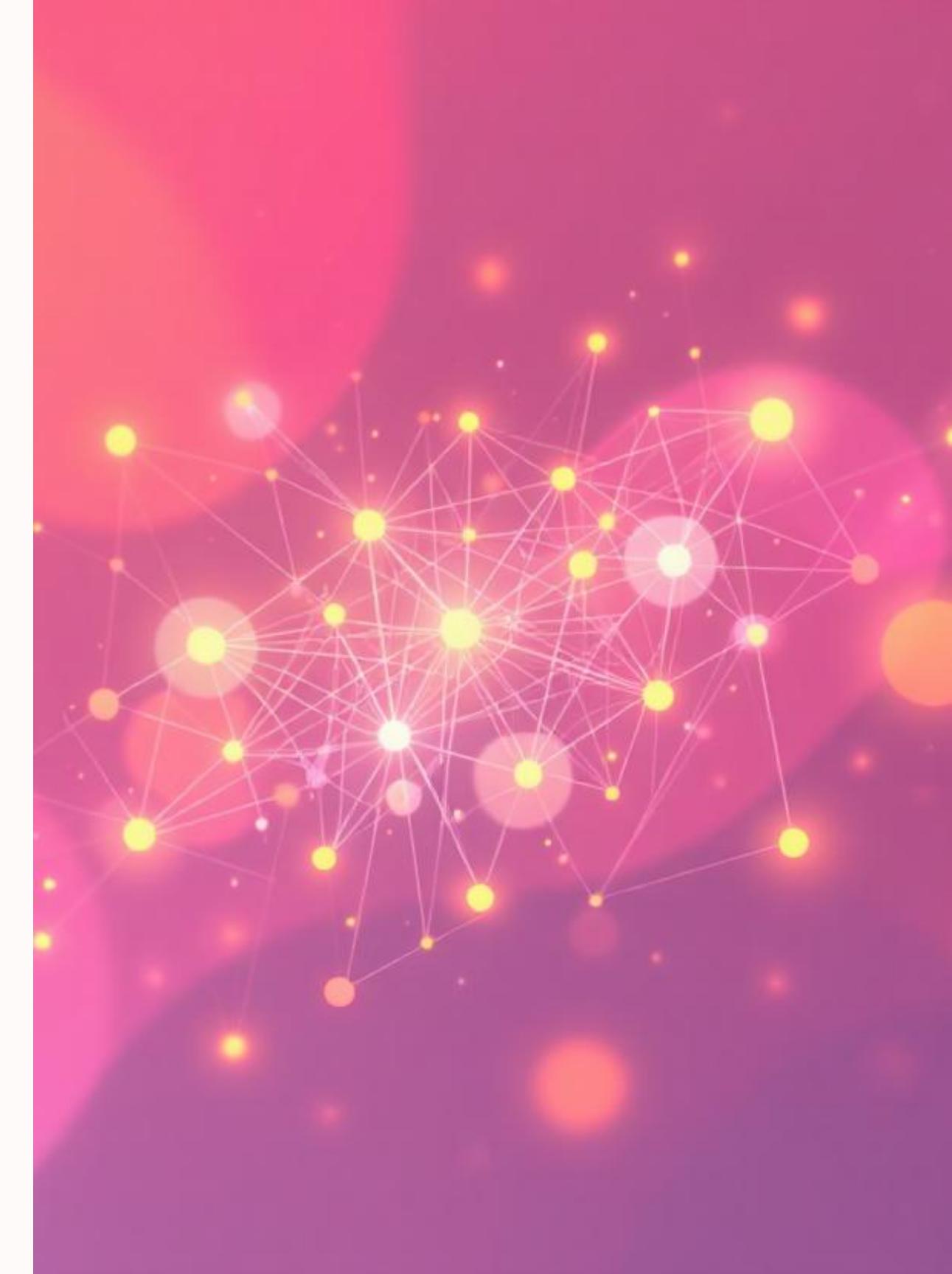
## *Sinergia para o Sucesso*

- Melhor resultado: sinergia entre arquitetura robusta e dataset rico
- Treinamento adequado é essencial



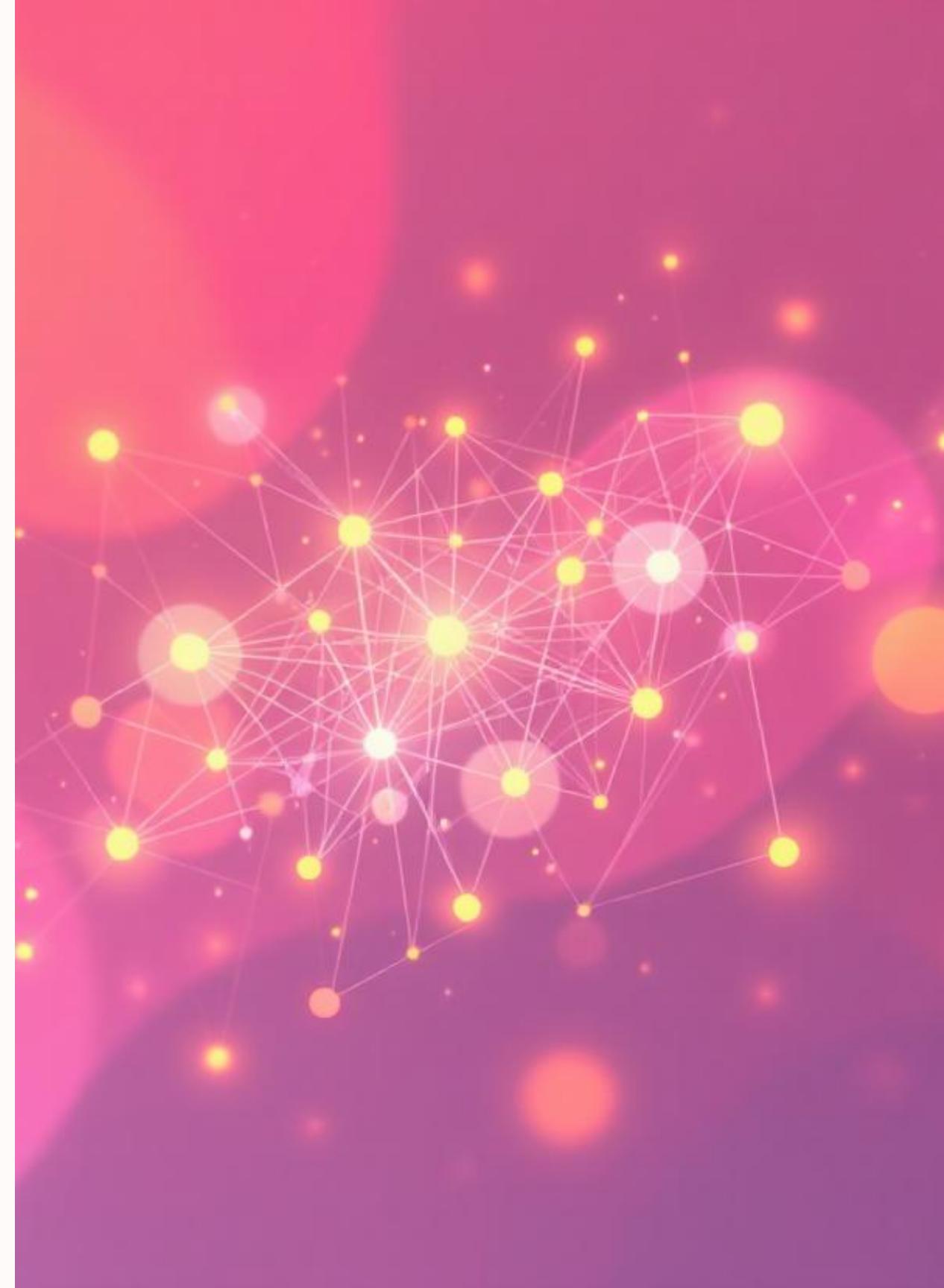
# Limitações

- Escopo limitado do painel de avaliação (5 pessoas).
- Foco exclusivo em piano solo, não generalizável para outros instrumentos.
- Geração determinística, sem explorar a "criatividade" com temperatura.



# Recomendações para Trabalhos Futuros

- Desenvolver métricas quantitativas de musicalidade para avaliação objetiva.
- Criar ferramentas interativas (plugins) para músicos, facilitando a experimentação.
- Explorar a geração de áudio bruto (WaveNet, Difusão) para maior realismo.



# Obrigado!

