

# DrumBeat

UMA ABORDAGEM MUSICAL PARA  
COMPILADORES

João Antônio Gomes Garcia

Insper | APS - Lógica da Computação



# Sumário

02



MOTIVAÇÃO: POR QUE CRIAR UMA LINGUAGEM DE BATERIA?

CARACTERÍSTICAS PRINCIPAIS: O QUE SUA LINGUAGEM PODE FAZER?

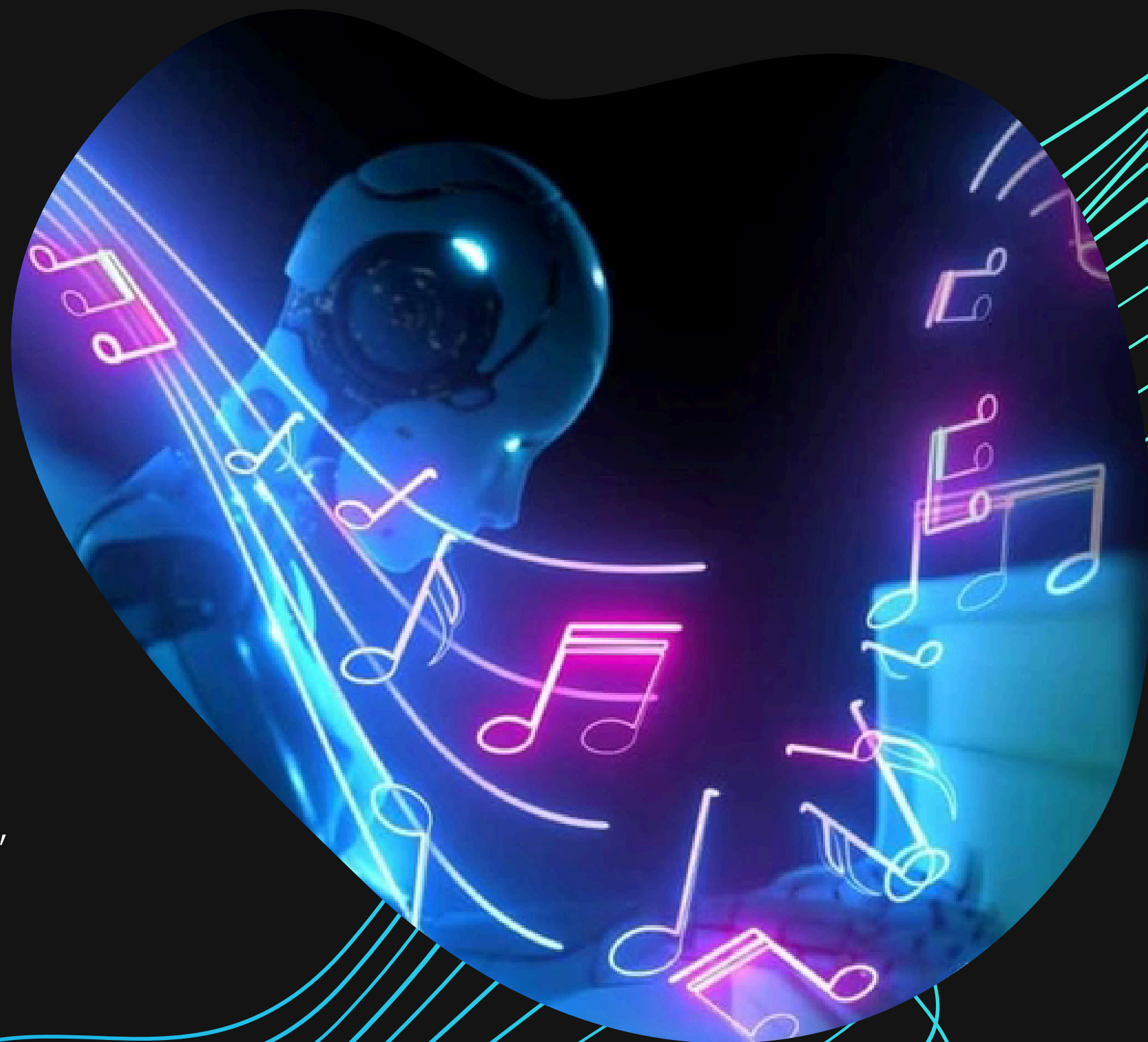
CURIOSIDADES TÉCNICAS: OS "BASTIDORES" DO PROJETO (FLEX, BISON, AST, VM).

EXEMPLOS DE CÓDIGO: DRUMBEAT EM AÇÃO.

DEMONSTRAÇÃO: OUVINDO A BATIDA!

# Por Que DrumBeat? A Visão por Trás da Linguagem

- Objetivo Acadêmico: Explorar as fases de um compilador (análise léxica, sintática, semântica, geração e execução de código) de forma prática e divertida.
- Paixão por Música: Unir o interesse por programação com a paixão pela bateria e ritmo.
- Facilitar a Criação Rítmica: Oferecer uma forma textual, simples e expressiva para músicos e programadores descreverem padrões de bateria, sem depender de softwares complexos ou interfaces gráficas.
- Abstração Musical: Como podemos traduzir conceitos musicais (tempo, instrumentos, repetições) para um modelo computacional?



# O Que DrumBeat Pode Fazer? Sintaxe e Funcionalidades

- Definição de Tempo (BPM): Controle a velocidade da sua batida.
  - Exemplo: bpm=120
- Instrumentos Simples: Caixa e Bumbo.
  - @ para caixa, # para bumbo.
- Modificadores de Duração/Ênfase: Ajuste o "feeling" da batida.
  - ! (rápido/curto), & (padrão), && (longo), &&& (muito longo).
- Silêncio (~): Crie pausas e espaços na sua batida.
- Repetições (xN): Loops de padrões rítmicos.
  - Exemplo: (#& @!)x2
- Agrupamento (+): Para tocar eventos em sequência dentro de um tempo
  - Exemplo: #+@
- Estrutura por Compassos (|): Organize sua música em blocos rítmicos.

# Curiosidades Técnicas (Análise Léxica e Sintática)

## OS BASTIDORES DE DRUMBEAT: COMO FUNCIONA?

- Análise Léxica (Flex):
  - O "ouvido" da linguagem: lê o código DrumBeat e o transforma em tokens (palavras-chave, símbolos, números).
  - Exemplo: bpm=120 vira BPM\_KEYWORD, IGUAL, NUMERO(120).
- Análise Sintática (Bison):
  - O "maestro" da linguagem: verifica se a sequência de tokens segue as regras gramaticais de DrumBeat.
  - Constrói a Árvore de Sintaxe Abstrata (AST): uma representação estruturada do seu programa, como um "esqueleto" da música.
  - A gramática foi definida em EBNF (Extended Backus-Naur Form).



# Curiosidades Técnicas (Interpretação: AST para VM)

## Os Bastidores de DrumBeat: Do Código à Batida!

- Fase de Compilação (Gerador de Bytecode):
  - Percorre a AST e a "traduz" para um conjunto de instruções de baixo nível, o bytecode.
  - Cada instrução é um OpCode (ex: OP\_PLAY\_KICK, OP\_LOAD\_BPM) com um operand (ex: índice do modificador, valor do BPM).
- Mini-VM (Máquina Virtual):
  - O "baterista eletrônico" da linguagem: lê e executa o bytecode, instrução por instrução.
  - Mantém o estado (BPM atual) e gerencia a pilha de repetições para loops aninhados.
  - Chama as funções de áudio (SDL\_mixer) para reproduzir os sons.
- Vantagens da VM: Modularidade, reuso, e uma forma de "interpretar" o código de forma estruturada.

# Exemplos de Código

## DRUMBEAT NA PRÁTICA: PADRÕES RÍTMICOS

```
1  {  
2  bpm=120  
3  @! #& |  
4  @! #& |  
5  }
```

```
1  {  
2  bpm=120  
3  @! #& |  
4  (#&)@! x2 |  
5  }
```

# Demonstração

HORA DA BATIDA! (DRUMBEAT EM AÇÃO)

[HTTPS://YOUTU.BE/PVR\\_9D\\_SMNC](https://youtu.be/PVR_9D_SMNC)



# Conclusão e Próximos Passos

## FLEXIBILIDADE

- A arquitetura com AST e VM permite fácil expansão da linguagem.

## POSSÍVEIS EVOLUÇÕES:

- Novos instrumentos (pratos, chimbal).
- Polifonia (tocar vários sons exatamente ao mesmo tempo).
- Funções mais avançadas de timing (swing, shuffle).
- Interface gráfica para compor.
- Exportação para MIDI.