# Linguagem Musical

Uma linguagem de programação com notações musicais

# Notações Utilizadas:

# **Notas e Pausas**

<b>)</b>	colcheia	if
	semicolcheia	else
-	Pausa de mínima	Fim de linha
7	Pausa de colcheia	comentário
	Pauta com clave de sol	input

# Acidentes e intervalos

8 <sup>va</sup>	Oitava à cima	multiplicação
$8^{vb}$	Oitava abaixo	divisão
Ь	bemol	subtração
#	sustenido	soma
<u> </u>	bequadro	igual

# Repetições e codas

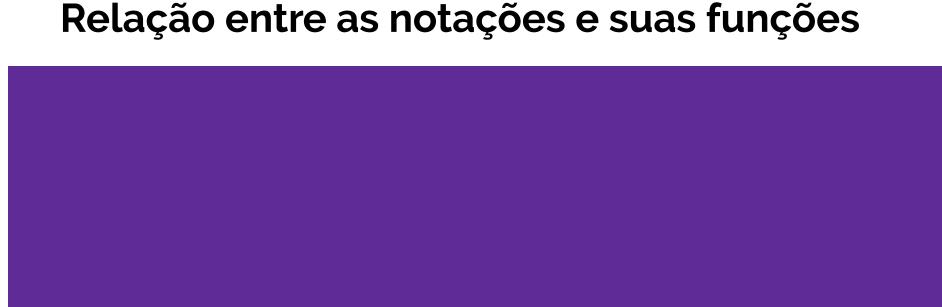
<b>%</b>	Repetição de compasso	igual comparativo
<b>+</b>	coda	return
%	segno	Definição de função
D.S.	Dal segno	Chamada de função
<u>  :</u> :	Barras de repetição	while

# Dinâmica e Ornamentos

tr	trinado	ou
2	grupetto	true
<b>∽</b>	Grupetto invertido	false
~	mordente	not
< > 	Crescendo e diminuendo	Menor e maior

# **Outros**

	cluster	and
<u></u>	microfone	print
,	respiraçao	Vírgula para separação de parâmetros
	Blocos de início e fim	Início e fim de bloco (funções)



# **Aritméticas**

## SOMA:

 O sustenido acrescenta na frequência da nota o equivalente a 1 semitom.

# SUBTRAÇÃO

- O bemol subtrai da frequência da nota um semitom.

## **IGUAL**:

O bequadro é usado para retirar um acidente.
 A nota tem seu valor "natural".

# MULTIPLUCAÇÃO:

A oitava acima tem o dobro da frequência da fundamental.

## DIVISÃO:

A oitava abaixo tem metade frequência da fundamental.

# Lógicas e Comparativas

## AND:

- O cluster é a união de muitas notas que não pertencem à mesma tonalidade.

## OR:

- O trilo diz que nota deve oscilar do início ao fim entre ela mesma e seu semitom acima.

## NOT:

 O mordente diz que nota deve uma vez, rapidamente subir pra o semitom a cima e voltar.

### **DIMINUENDO:**

 O diminuendo diz que a dinâmica daquele pedaço da música deve diminuir gradativamente.

#### **CRESCENDO:**

O crescendo diz que a dinâmica daquele pedaço da música deve aumentar gradativamente.

#### **IGUAL COMPARATIVO:**

A repetição de compasso diz que o próximo compasso deve ser igual ao que está sendo tocado no momento.

# Funções, while e if

# DEFINIÇÃO DE FUNÇÃO:

- O segno marca um pedaço da música que deve ser tocado sempre que necessário.

# CHAMADA DE FUNÇÃO:

- O dal segno indica que o segno deve ser tocado.

### **RETURN:**

 A coda costuma ser a parte final da música e é chamada quando todas as outras partes já foram tocadas.

## IF:

A colcheia é uma nota que dura um oitavo do "tempo"

### ELSE:

- A semicolcheia é uma nota que dura 1/16 do "tempo"

### WHILE:

- As barras definem o início e o fim do bloco que deve ser repetido.

# **Outros**

## PRINT:

- O microfone capta o som e, se ligado a um amplificador "tira a música da partitura"

## **INPUT:**

 Na pauta a música deixa de ser passada oralmente e é registrada. Sai do interlocutor e se transforma e código.

## VÍRGULA:

 A respiração é marcada para que o músico dê espaço entre as notas tocadas. Ela dá naturalidade para as frases musicais.

# FIM E INÍCIO DE BLOCO DE FUNÇÃO:

- As barras delimitam o fim e o início da música. Nesse caso seria de cada parte da música. Geralmente as barras duplas simples seriam utilizadas, mas nesse caso pareceu fazer mais sentido enfatizar o início e o fim.

#### FINAL DE LINHA:

- A pausa de mínima dura dois tempo.

## **COMENTÁRIOS:**

- A pausa de colcheia dura 1/8 de tempo.

# **EBNF**

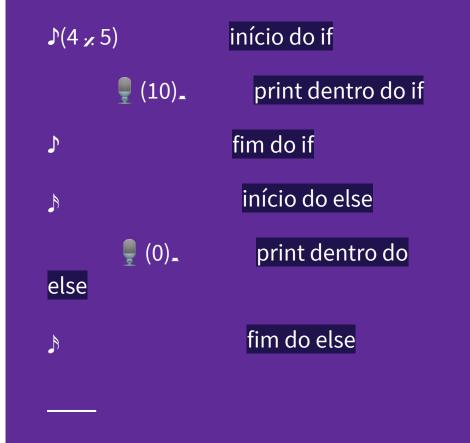
```
PROGRAM = { COMMAND };
COMMAND = ((\lambda \mid ASSIGNMENT \mid PRINT \mid FUNCCALL \mid RETURN), "_ ") \mid WHILE \mid IF \mid FUNCASSIG \mid COMMENT;
ASSIGNMENT = IDENTIFIER, "\( \bar{\pi} \), RELEXP;
PRINT = " | ", EXPRESSION;
WHILE = "|", "(", RELEXP, ")", CODE, "|";
IF = ("♪", "(", RELEXP, ")", BLOCK, "♪") | ("♪", "(", COND, ")", BLOCK, "♪", "♭", CODE, "♭");
FUNCASSIG = "*", "|", VAR, "(", PARAMASSIG, ")", BLOCK, "|";
PARAMCALL = [ RELEXP { "," RELEXP } ];
PARAMASSIG = [IDENTIFIER { "," IDENTIFIER } ];
RETURN = ", [RELEXP], ", "
RELEXP = EXPRESSION, \{("_{\varkappa}" \mid "_{\neg}" \mid "_{\neg}"), \text{EXPRESSION}\};
EXPRESSION = TERM, \{("\sharp" \mid "\flat" \mid "_{\sharp}"), TERM \};
TERM = FACTOR, { ("" | "" | "" | "] "), FACTOR };
FACTOR = (("\sharp" \mid " \mid " \mid ","), FACTOR) \mid NUMBER \mid (as identifier "(" PARAMCALL ")") \mid "(", RELEXP, ")" \mid IDENTIFIER \mid INPUT;
INPUT = "♣";
```

# **EBNF**

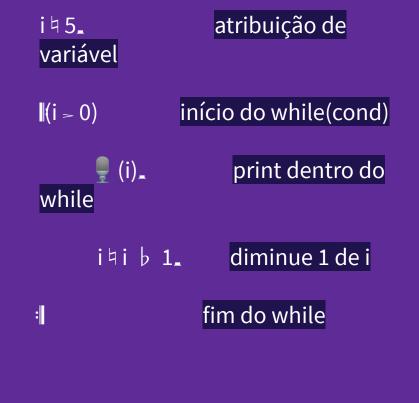
```
IDENTIFIER = LETTER, { LETTER | DIGIT | "_" };
NUMBER = DIGIT, { DIGIT };
LETTER = (a | ... | z | A | ... | Z );
DIGIT = (1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 );
COMMENT = ", ", {*};
```

# Exemplos

# If e else



# while



# Print, input e comentário



input para variável



print da variável

, comentário, comentário

# função

