### Linguagem Musical

Uma linguagem de programação com notações musicais

# Notações Utilizadas:

#### **Notas e Pausas**

Notações utilizadas

- → colcheia
- > semicolcheia
- Pausa de mínima
- , Pausa de colcheia



# Acidentes e intervalos

Notações utilizadas

8va - oitava acima

8<sup>vb</sup> - oitava abaixo

- bemol

# - sustenido

ا - bequadro

# Repetições e codas

Notações utilizadas

- repetição de compasso
- + coda
- 🐒 segno
- barras de início e fim de repetição

D.S. - dal segno

### Dinâmica e Ornamentos

Notações utilizadas

tr - trilo

→ grupetto

- grupetto invertido

- mordente

crescendo e diminuendo

#### **Outros**

Notações utilizadas

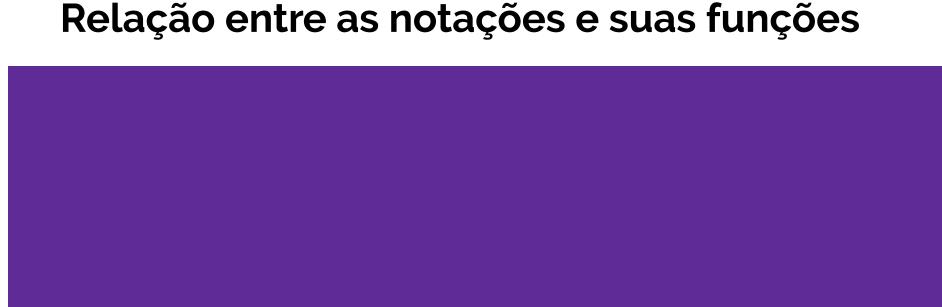




- microfone

- respiração





#### **Aritméticas**

A soma é representada pelo sustenido.

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

A subtração é representada pelo bemol.

O bemol subtrai da frequência da nota um semitom.

O igual é representado pelo bequadro

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

A multiplicação é representada pelo intervalo de oitava acima.

A oitava acima tem o dobro da frequência da fundamental.

A divisão é representada pelo intervalo de oitava abaixo.

A oitava abaixo tem metade frequência da fundamental.

#### Lógicas e Comparativas

O and é representado pelo cluster.

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

O or é representado pelo trilo.

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

O not é representado pelo mordente.

 O mordente diz que nota deve uma vez, rapidamente subir pra o semitom a cima e voltar. O maior é representado pelo diminuendo.

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

O menor é representado pelo crescendo.

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

O igual comparativo é representado pela repetição de compasso.

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

#### Funções, while e if

A definição de função é representada pelo segno.

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

A chamada de função é representada pelo dal segno.

 O dal segno indica que o segno deve ser tocado.

O return é representado pela coda.

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

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

O else é representado pela semicolcheia.

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

O while é representado pelas barras de início e fim da repetição

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

#### **Outros**

O print é definido pelo microfone.

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

O input é a pauta com clave de sol.

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

A vírgula é representada pela respiração.

 A respiração é marcada para que o músico dê espaço entre as notas tocadas. Ela dá naturalidade para as frases musicais. Os blocos de cada função são representados pelas barras de início e fim.

- 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.

O final de linha é representado pela pausa de mínima.

- A pausa de mínima dura dois tempo.

Os comentário são representados pela pausa de colcheia

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

#### **EBNF**

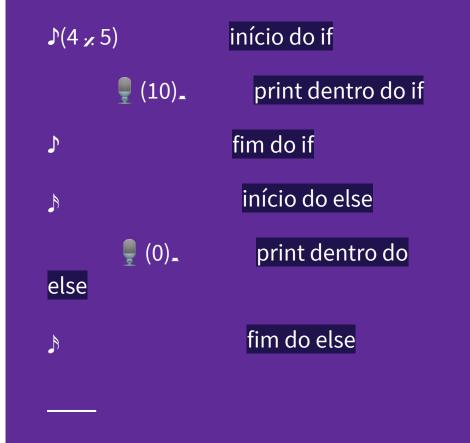
```
PROGRAM = { COMMAND };
COMMAND = ((\lambda \mid ASSIGNMENT \mid PRINT \mid FUNCCALL \mid RETURN), "_ ") \mid WHILE \mid IF \mid FUNCASSIG \mid COMMENT;
ASSIGNMENT = IDENTIFIER, "\( \bar{\pi} \), RELEXP;
PRINT = " | ", EXPRESSION;
WHILE = "|", "(", RELEXP, ")", CODE, "|";
IF = ("♪", "(", RELEXP, ")", 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 (_{\&} IDENTIFIER "(" PARAMCALL ")") \mid "(", RELEXP, ")" \mid IDENTIFIER \mid INPUT;
INPUT = "♣";
```

#### **EBNF**

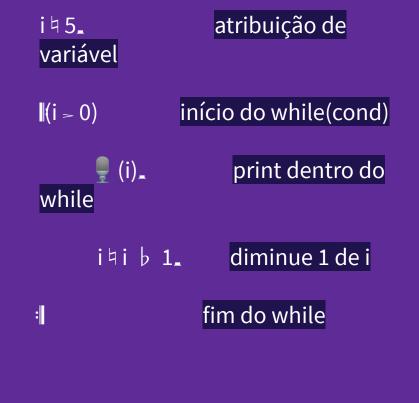
```
IDENTIFIER = LETTER, { LETTER | DIGIT | "_" };
NUMBER = DIGIT, { DIGIT };
LETTER = (a | ... | z | A | ... | Z );
DIGIT = (1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 );
COMMENT = ", ", {*};
```

### Exemplos

#### If e else



#### while



# Print, input e comentário



input para variável



print da variável

, comentário, comentário

## função

