

Gramática LL(1)

- Exemplo de programa **aceito** pela BNF descrita neste arquivo:

```
variavel_inteira = 5
variavel_booleana = verdadeiro

funcao imprimir_ola(){
    imprimir("Ola, Mundo!")
}

funcao somar(a, b){
    retornar a + b
}

resultado = 0

imprimir_ola()

resultado = somar(3, 4)

se (resultado > 10){
    imprimir("Resultado e maior que 10.")
}
senao{
    imprimir("Resultado e 10 ou menor.")
}

contador = 0
enquanto (contador < resultado){
    imprimir("Contando: {contador}")
    contador += 1
}
se (contador == 5){
    imprimir("Pulando 5")
    continuar
}
se (contador == 7){
    imprimir("Parando no 7")
}
```

```

        interromper
    }

    MINHA_CONSTANTE = "Esta e uma constante"
    imprimir(MINHA_CONSTANTE)

    imprimir("Variável booleana: {variavel_booleana}")

    resultado_aritmetico = (variavel_inteira + resultado) - 5 * 2 / 2

    se (resultado_aritmetico == 10){
        imprimir("Alguma condicao booleana foi atendida.")
    }

```

- Exemplo de programa **não** aceito pela BNF descrita neste arquivo:

```

variavel_inteira = 5
variavel_booleana = True

funcao imprimir_ola(){
    imprima("Ola, Mundo!") # Erro: 'imprima' não é uma palavra-chave válida, deve
    ser 'imprimir'
}

funcao somar(a, b){
    retorne a + b # Erro: 'retorne' não é uma palavra-chave válida, deve ser 'retornar'
}

resultado = 0

imprimir_ola()

resultado = somar(3, 4)

se (resultado > 10){
    imprimir("Resultado e maior que 10.")
} senao{
    imprimir("Resultado e 10 ou menor.")
}

contador = 0

enquanto (contador < resultado){
    imprimir(f"Contando: {contador}")
    contador += 1
    se (contador == 5){
        imprimir("Pulando 5")
    }
}

```

```

    seguindo # Erro: 'seguindo' não é uma palavra-chave válida, deve ser
    'continuar'
}

se (contador == 7){
    imprimir("Parando no 7")
    parar # Erro: 'parar' não é uma palavra-chave válida, deve ser 'interromper'
}

MINHA_CONSTANTE = "Esta e uma constante"
imprimir(MINHA_CONSTANTE)

imprimir(f"Variavel booleana: {variavel_booleana}")

resultado_aritmetico = (variavel_inteira + resultado) - 5 * 2 / 2

se (resultado_aritmetico == 10){
    imprimir("Alguma condicao booleana foi atendida.")
}

```

Erros:

- `imprima("Ola, Mundo!")` # Erro: 'imprima' não é uma palavra-chave válida, deve ser 'imprimir'
- `retorne a + b` # Erro: 'retorne' não é uma palavra-chave válida, deve ser 'retornar'
- `seguindo` # Erro: 'seguindo' não é uma palavra-chave válida, deve ser 'continuar'
- `parar` # Erro: 'parar' não é uma palavra-chave válida, deve ser 'interromper'

Descrição BNF da Linguagem

`<programa> ::= <declaracao_variavel> {<declaracao_funcao>} <comandos>`

`<declaracao_variavel> ::= <identificador> = <valor>`

`<declaracao_funcao> ::= funcao <identificador> (<parametros>) -> <tipo_retorno>:
<comandos_funcao>`

`<parametros> ::= <parametro> {, <parametro>}`

`<parametro> ::= <identificador>`

`<tipo_retorno> ::= inteiro | booleano | vazio`

`<comandos> ::= <comando> {<comando>}`

`<comandos_funcao> ::= <comando> {<comando>}`

<comando> ::= <declaracao_variavel>
| <chamada_funcao>
| se <expressao>: <comandos> senao: <comandos>
| enquanto <expressao>: <comandos>
| interromper
| continuar
| imprimir(<expressao>)
| <condicao_especial>

<chamada_funcao> ::= <identificador> (<argumentos>)

<argumentos> ::= <expressao> {, <expressao>}

<condicao_especial> ::= se <expressao> == <valor>: <comandos> senao: <comandos>

<expressao> ::= <termo> {<op_aditivo> <termo>}

<termo> ::= <fator> {<op_multiplicativo> <fator>}

<fator> ::= <identificador>
| <numero>
| <booleano>
| (<expressao>)

<op_aditivo> ::= + | -

<op_multiplicativo> ::= * | /

<identificador> ::= <letra> {<letra> | <digito> | _}

<letra> ::= a|b|...|z|A|B|...|Z

<digito> ::= 0|1|...|9

<numero> ::= <digito> {<digito>}

<booleano> ::= verdadeiro | falso