



# COMPILADORES FOOD

MATÉRIA: CONSTRUÇÃO DE COMPILADORES

AUTORES: FELIPE HENRIQUE, KAIQUE ANDRADE



**01** CONCEITO ESCOLHIDO

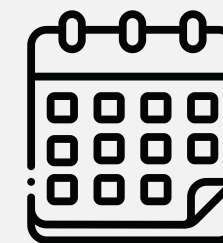
**02** CONTEXTUALIZAÇÃO

**03** DESCRIÇÃO BNF

**04** PROGRAMA ACEITO

**04** PROGRAMA INVÁLIDO

**04** TRATAMENTO DE ERROS



**ETAPAS PREVISTAS**

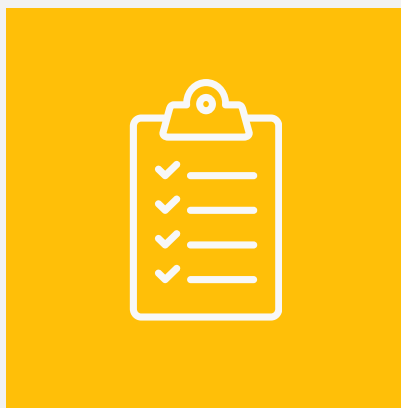


## CONCEITO

Comida é o que está no nosso cotidiano possuindo muitas palavras fáceis de compreensão, nosso compilador vai montar uma estrutura básica com palavras de alimentos ou derivados da culinária brasileira.

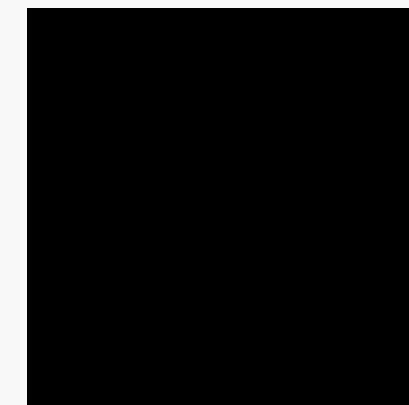


# CONCEITO

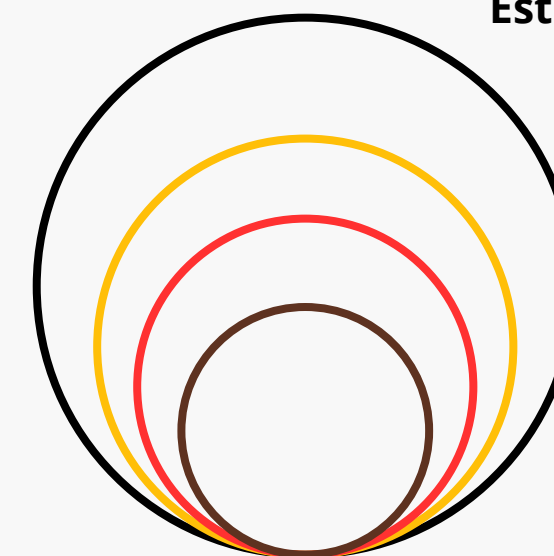


## ESTRUTURA

Estrutura regular com tipagem de dados  
abertura e fechamento de blocos e aplicação  
de condicionais básicas



## GRAMÁTICA REGULAR



Estrutura de fase

**sensível ao contexto**

**livre de contexto**  
regular



## ESTRUTURA DO PROGRAMA

<INICIOPROGRAMA> ::= prato  
<ABREBLOCO> ::= {  
<FECHABLOCO> ::= }  
<ABREBLOCOND> ::= (  
<FECHABLOCOND> ::= )  
< FIM\_STRUC> ::= .

## TIPAGEM DOS DADOS

<TPINTEIRO> ::= torrada  
<TPSTRING> ::= macarronada  
<TIPOBOOL> ::= processado  
<TPREAL> ::= agua

## OPERADORES LÓGICOS E ARITIMETICOS

<MAIS> ::= +  
<MENOS> ::= -  
<MULTI> ::= \*  
<DIV> ::= /  
<MAIOR> ::= >  
<MENOR> ::= <  
<ATRIBUI> ::= =

## ESTRUTURA CONDICIONAIS

<CONDIF> ::= almoco  
<CONDELSE> ::= sobremesa  
<REPWHILE> ::= rodizio  
<REPFOR> ::= degustacao  
<ENTAO> ::= coma

## TOKENS IDs

< NUMERO> ::= (< DIGITO >)+  
< #DIGITO> ::= [ 0-9 ] < ID> ::=  
<LETRA> (< LETRA> | < DIGITO>)\*  
< LETRA> ::= [A-Z] | [a-z]  
< BOOL> ::= (TRUE | FALSE | true | false)  
< ASPAS> ::= '

# PROGRAMA

## ESTRUTURA BÁSICA

<INICIOPROGRAMA> <ID>

<ABREBLOCO>

<declara\_variavel>

<body>

<FECHABLOCO>

## RESULTADO

```
prato teste {
```

```
    torrada pao.
```

```
        torrada quantidadePessoa = 3.
```

```
        torrada quantidadeLugares = 4.
```

```
        processado podeServir = 'false'.
```

```
        processado limparMesa = 'false'.
```

```
    almoco quantidadePessoa <= quantidadeLugares
```

```
    coma (
```

```
        podeServir = 'true'
```

```
    ) sobremesa (
```

```
        limparMesa = 'true'
```

```
    )
```

```
}
```

# EXEMPLO DE UM PROGRAMA VALIDO

```
Food [Java Application] /home/desbravador/Desktop/SSD/kaique/Downloa
peça sua comida
esperando pedido: prato teste
{
    torrada pao.
    torrada quantidadePessoa = 3.
    torrada quantidadeLugares = 4.
    processado podeServir = 'false'.
    processado limparMesa = 'false'.

    almoco quantidadePessoa <= quantidadeLugares coma (
        podeServir = 'true'
    ) sobremesa (
        limparMesa = 'true'
    )
}
COMIDA ENTREGUE
```

**DECLARAÇÃO DE VARIÁVEIS**

**CONDICIONAL**  
VERIFICA SE QUANTIDADE DE PESSOA É MENOR QUE QUANTIDDE DE LUGARES DISPONÍVEIS

**AÇÃO**  
DECIDE SE SERVE OU SE LIMPA OS LUGARES



# EXEMPLO DE UM PROGRAMA INVALIDO

```
Problems | Javadoc | Declaration | Console x | JavaCC Console | Coverage
<terminated> Food [Java Application] /home/desbravador/Desktop/SSD/kaique/Down
prato restaurante_kaique
{
    torrada pao.
    torrada quantidadePessoa = 3.
    torrada quantidadeLugares = 4.
    processado podeServir = 'false'.
    processado limparMesa = 'false'.

    almoco quantidadePessoa <= quantidadeLugares coma (
        podeServir = 'true'
    ) sobremesa (
        limparMesa = 'true'
    )
}Oops.
Linha com erro lexico 15, column 18. possível erro: '95' (95),
```

CARÁCTER NÃO RECONHECIDO

TRATAMENTO DE ERRO



# OBRIGADO

**<https://github.com/Ilustra/compiladorfood>**