

## Roteiro 02 – Júlia e Marcella

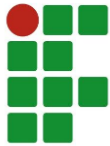
### Atividade 1, saída:

```
marce@DESKTOP-KCMR5HR MINGW64 ~/Documents/IF/Compiladores/SEMANA2/Roteiro2/Marcella_Julia/atv1
$ java Roteiro02 teste.txt
line: 1 col: 1 match: --abc123--
action [28] { return new Symbol(sym.IDENT, yyline, yycolumn, yytext()); }
#6
line: 1 col: 7 match: --
--
action [26] { return new Symbol(sym.ESPACO, yyline, yycolumn, yytext()); }
#9
line: 2 col: 1 match: --123--
action [30] { int aux = Integer.parseInt (yytext());
              return new Symbol(sym.INTEIRO, yyline, yycolumn, new Integer(aux)); }
#7
line: 2 col: 4 match: --abcd--
action [28] { return new Symbol(sym.IDENT, yyline, yycolumn, yytext()); }
#6
line: 2 col: 8 match: --
--
action [26] { return new Symbol(sym.ESPACO, yyline, yycolumn, yytext()); }
#9
line: 3 col: 1 match: --// linha comentada--
action [24] { return new Symbol(sym.COMENTLINE, yyline, yycolumn, yytext()); }
#8
#-1
```

Figura 1: “java Roteiro02 teste.txt”.

### Atividade 1: Note que a Figura 4 será um pouco diferente. Em que? Porque?

O resultado ficou diferente no que se refere aos “#” que indicam a saída do estado que a palavra casou. E também ocorreram mais match’s, referentes aos espaços ou fim de linha; e comentários. A saída do código é diferente pois agora alteramos o valor de identificação das variáveis e também acrescentamos o estado de espaços e comentários.



## Atividade 2, saída:

```
MINGW64:/d/Compiladores/Marcella_Julia/atv2
julia@DESKTOP-ULDH4P2 MINGW64 /d/Compiladores/Marcella_Julia/atv2 (master)
$ java Main
1+4+5;
2*5;
10/5;
8-4;
```

**Figura 2.** “java Main”.

Modificamos os arquivos .cup e .flex para reconhecer também operações de divisão e multiplicação como mostra na Figura 2.