

# Gerador de tabela Verdade

## Grupo:

- Rodrigo Junger de Carvalho
- Roberto Silva Lourenço
- Eriky Nunes

## Atuações:

O trabalho foi feito em reuniões semanais por vídeo conferencia. No final do trabalho, a criação dos códigos ficaram da seguinte maneira:

- Rodrigo: elaborou e criou as funções calculaLinha e interpretaExpressao.
- Rodrigo e Roberto: percebemos que poderíamos reaproveitar o raciocínio feito na função calculaLinha para extrair as sub-expressões da expressão principal, com isso criamos a gera geraSubStrings
- Eriky: ficou responsável por fazer a exibição da tabela verdade junto com a função (denominacaoFuncao) que informa se a formula é uma Tautologia, Satisfatória e Insatisfatória.
- Rodrigo, Roberto e Eriky: Ficaram responsáveis por debugar o código e fazer testes.

## Roteiro completo de compilação/execução

Link Youtube: <https://youtu.be/pQQEJRZuwKo>

Link GitHub: <https://github.com/robertosl/geradortabelaverdade>

1. Verifique se você possui haskell instalado na sua máquina. Você pode usar os códigos abaixo para verificar: GHC's repl:

```
ghci --version
The Glorious Glasgow Haskell Compilation System, version 8.8.4
```

obs: Caso não tenha, faça o download. Recomendamos uma das versões abaixo:

- <https://www.haskell.org/downloads/#stack>
- <https://www.haskell.org/downloads/#platform>

2. Com o Haskell já instalado na sua máquina agora abra o seu terminal
3. Abra a pasta do projeto pelo terminal
4. Digite no terminal

```
ghci
```

Provavelmente agora o ghci vai estar rodando e bem provável que abaixo apareça:

```
Prelude>
```

Isso quer dizer que o Prelude está rodando

5. Após, na frente do "Prelude>" digite:

```
:l geradorTabelaVerdade.hs
```

Isso vai compilar o código do gerador. Provavelmente você receberá a seguinte mensagem:

```
[1 of 1] Compiling Main                ( geradorTabelaVerdade.hs, interpreted )
Ok, one module loaded.
```

E ao invés do "Prelude>" agora aparecerá "\*Main>"

6. Agora basta você digitar

```
main
```

OBS: é necessário escrever o "main" sempre para escrever uma fórmula Agora basta que você insira a fórmula desejada. Abaixo vamos mostrar alguns exemplos e observações sobre o seu input.

7. Caso deseje sair do ghci basta digitar

```
:q
```

## Exemplos de entrada e Saída

Antes de mostrarmos os exemplos de entrada e saída gostaria de salientar alguns cuidados que você deve ter na hora de colocar sua entrada:

### Operadores:

- "&": e
- "|": ou
- "~": negação
- "->": se, então

- "<->": se, somente se

## Cuidados

1. Use a notação pré-fixada
2. É necessário colcar espaço após cada operador e variável

Exemplo válido:  $\sim \rightarrow a b$  Exemplo não válido:  $\sim \rightarrow ab$

## Exemplos

```
-> a b
"a | b | -> a b"
"V | V | V"
"V | F | F"
"F | V | V"
"F | F | V"
"Formula satisfativel"
```

```
<-> -> & a b c -> a -> b c
"a | b | c | & a b | -> b c | -> & a b c | -> a -> b c | <-> -> & a b c -> a
-> b c"
"V | V | V | V | V | V | V | V"
"V | V | F | V | F | F | F | V"
"V | F | V | F | V | V | V | V"
"V | F | F | F | V | V | V | V"
"F | V | V | F | V | V | V | V"
"F | V | F | F | F | V | V | V"
"F | F | V | F | V | V | V | V"
"F | F | F | F | V | V | V | V"
"Tautologia"
```

```
& a ~ a
"a | ~ a | & a ~ a"
"V | F | F"
"F | V | F"
"Formula insatisfativel"
```