

# RELATÓRIO SOBRE PROPOSIÇÕES LÓGICAS E SUA REPRESENTAÇÃO EM LÓGICA COMPUTACIONAL

## 1 INTRODUÇÃO

Este relatório tem como objetivo apresentar o conceito de **proposição lógica** e sua fundamental importância, tanto nos estudos da lógica formal quanto em sua aplicação no campo da lógica computacional. Serão abordadas as características de uma proposição, sua natureza binária de valor de verdade e como ela é representada em linguagens de programação, especificamente em Python. O estudo da lógica permite entender como o raciocínio é formado.

## 2 DESENVOLVIMENTO

### 2.1 Conceito e Natureza da Proposição Lógica.

Uma **proposição** é definida como um **enunciado ou uma frase declarativa. pode ser classificada como verdadeira ou falsa**, mas **nunca ambas ao mesmo tempo**. ela **garante que, com base em premissas verdadeiras, seja possível chegar a conclusões verdadeiras**.

A lógica clássica, em particular, é regida por três princípios fundamentais que sublinham essa natureza binária das proposições:

- **Princípio da Identidade:** Toda proposição é idêntica a si mesma ( $P \text{ é } P$ ).
- **Princípio da Não Contradição:** Uma proposição não pode ser verdadeira e falsa ao mesmo tempo (não ( $P \text{ e não } P$ )).
- **Princípio do Terceiro Excluído:** Toda proposição ou é verdadeira ou é falsa, não havendo uma terceira possibilidade para sua valoração ( $P \text{ ou não } P$ ).

2.2 Representação de Proposições em Python, a representação mais direta e fundamental de uma proposição lógica e seu respectivo valor de verdade é feita por meio do **tipo de dado booleano (bool)**. Este tipo de dado só pode assumir dois valores: **True (Verdadeiro)** ou **False (Falso)** [mencionado na conversa anterior]. Esta característica alinha-se perfeitamente com a natureza binária da lógica clássica (1 para Verdadeiro e 0 para Falso).

No exemplo prático fornecido pelo Google Colab:

- Tipo de dado de proposicao\_1: <class 'bool'>
- Tipo de dado de proposicao\_2: <class 'bool'>
- Tipo de dado de proposicao\_3: <class 'bool'> Esta saída confirma que as variáveis que armazenam os valores de verdade das proposições são do tipo **bool**, demonstrando a representação computacional dos conceitos lógicos.

Os valores atribuídos às proposições:

- **Proposição 1: True** ("O Brasil é um país da América Latina.")
- **Proposição 2: False** ("Minas Gerais é um estado do Nordeste.")
- **Proposição 3: True** ("Três mais um é igual a quatro.") Ilustram como proposições são expressas e seus valores de verdade explicitamente definidos ou avaliados em Python. No caso da "Proposição 3", a expressão **(3 + 1 == 4)** é avaliada pelo Python, que retorna automaticamente **True** porque a igualdade é satisfeita. Isso exemplifica como operações de

comparação e matemáticas são intrínsecas à determinação de valores de verdade em contextos computacionais, sendo a base da **lógica simbólica (matemática)** que originou a lógica utilizada em computadores.

### 2.3 Tipos de Proposições e Conectivos Lógicos

As proposições podem ser classificadas como **simples** (quando contêm uma única afirmação) ou **compostas** (quando são formadas por duas ou mais proposições simples "ligadas" por um **conectivo lógico**). Os principais conectivos lógicos incluem:

- **Conjunção** (**e**, **AND**, **∧**): Resulta em verdadeiro somente se todas as proposições simples conectadas forem verdadeiras.
- **Disjunção** (**ou**, **OR**, **∨**): Na forma inclusiva, resulta em falso somente se todas as proposições simples conectadas forem falsas.
- **Negação** (**não**, **NOT**, **¬**): Opera sobre uma única proposição, invertendo seu valor de verdade (Vira Falso se for Verdadeiro, e Verdadeiro se for Falso).

Esses conectivos são essenciais para construir expressões lógicas mais complexas e são amplamente utilizados em **algoritmos e estruturas de decisão** em programação, onde condições booleanas (que resultam em **True** ou **False**) são fundamentais para controlar o fluxo de execução de um programa

## 3 CONCLUSÃO

A proposição lógica é a unidade fundamental do raciocínio, definida como uma frase declarativa que é inequivocamente verdadeira ou falsa. Sua representação em Python através do tipo booleano (**bool**) com os valores **True** e **False** demonstra a aplicação direta dos princípios da lógica formal no contexto da computação. A capacidade de manipular e combinar essas proposições por meio de conectivos lógicos é a base para a construção de algoritmos complexos e a compreensão de linguagens de programação, tornando a lógica um pilar indispensável no desenvolvimento de sistemas computacionais.

## REFERÊNCIAS BIBLIOGRÁFICAS

MIRANDA, Eduardo Furlan. **Fundamentos de lógica**. Material didático (LMC-11-Fundamentos\_2024-08-21.pdf). 2024.

SCHEFFER, Vanessa Cadan; VIEIRA, Gilberto; LIMA, Thiago Pinheiro Félix da Silva e. **Lógica Computacional**. Londrina: Editora e Distribuidora Educacional S.A., 2020.