



**INSTITUTO FEDERAL**

Rio Grande do Norte

Campus Lajes

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO  
GRANDE DO NORTE  
CAMPUS AVANÇADO LAJES  
CURSO TÉCNICO INTEGRADO EM IFORMÁTICA

LUIZ FELIPE DAS CHAGAS BEZERRA

**RELATÓRIO DE PRÁTICA PROFISSIONAL: MONITORIA DE FUNDAMENTOS DE  
LÓGICA E ALGORITIMO**

LAJES/RN  
2023

LUIZ FELIPE DAS CHAGAS BEZERRA

**RELATÓRIO DE PRÁTICA PROFISSIONAL: MONITORIA DE FUNDAMENTOS DE  
LÓGICA E ALGORITIMO**

Relatório de Prática Profissional apresentado ao Curso Técnico Integrado em Informática do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Campus Avançado Lajes, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática.

Orientador (a): Prof<sup>o</sup> Me. Dannilo Martins Cunha.



**INSTITUTO FEDERAL**

Rio Grande do Norte

Campus Lajes

## **RESUMO**

A prática profissional em questão foi caracterizada na forma de uma monitoria, que teve como objetivo auxiliar os iniciados no curso de Informática, tendo em vista que a disciplina de Fundamentos de Lógica e Algoritmo é a base para os jovens na área da programação. As atividades realizadas serviram principalmente para o enriquecimento curricular dos monitores voluntários, uma vez que executaram atividades relacionadas ao âmbito acadêmico. A presente monitoria foi realizada de maneira presencial. Os objetivos foram em parte atingidos, uma vez que permitiu aos monitores se desenvolverem como profissionais, mas a ausência dos iniciados em informática é uma falha presente no desenvolvimento desta atividade de prática profissional. É válido ressaltar que tal ausência era algo esperado devido ao fato que os iniciados não costumam procurar os monitores, mas ainda assim a prática agregou aos monitores responsáveis. Por fim os resultados obtidos foram o aprimoramento dos conhecimentos acadêmicos, ganho de experiência com as ferramentas em questão e comprometimento.

Palavras-chave: Monitoria. Lógica. Algoritmos. Prática profissional.



## LISTA DE ILUSTRAÇÕES

<b>Figura 1 – Proposições simples.</b>	<b>11</b>
<b>Figura 2 – Proposições compostas.</b>	<b>11</b>
<b>Figura 3 – Exemplificação do método de determinar linhas.</b>	<b>14</b>
<b>Figura 4 – Redução de conectivos (Simplificação de implicação)</b>	<b>20</b>
<b>Figura 5 – Algoritmo para multiplicar dois valores (Descrição Narrativa)</b>	<b>20</b>
<b>Figura 6 – Algoritmo para descobrir a média final ponderada (Fluxograma).</b>	<b>21</b>
<b>Figura 7 – Algoritmo para saber a média final comum (pseudocódigo)</b>	<b>21</b>
<b>Figura 8 – Algoritmo de verificação de idade no Portugol (pseudocódigo)</b>	<b>25</b>
<b>Figura 9 – Algoritmo para montar tabuada no Portugol (Pseudocódigo)</b>	<b>26</b>
<b>Figura 10 – Tela de início do Portugol Studio.</b>	<b>31</b>



## LISTA DE TABELAS

<b>Tabela 1 – Síntese das atividades do aluno na monitoria.</b>	<b>9</b>
<b>Tabela 2 – Exemplo de tabela-verdade.</b>	<b>14</b>
<b>Tabela 3 – Exemplificação da proposição <math>s \wedge q</math>.</b>	<b>14</b>
<b>Tabela 4 – Tabela verdade da proposição <math>(p \wedge q) \vee r</math>.</b>	<b>15</b>
<b>Tabela 5 – Tabela verdade de uma tautologia.</b>	<b>15</b>
<b>Tabela 6 – Tabela verdade de uma contradição.</b>	<b>16</b>
<b>Tabela 7 – Tabela verdade de uma contingência (Negação)</b>	<b>16</b>
<b>Tabela 8 – Operadores aritméticos.</b>	<b>23</b>
<b>Tabela 9 – Operadores lógicos.</b>	<b>23</b>
<b>Tabela 10 – Operadores relacionais.</b>	<b>24</b>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
1.1	CONTEXTUALIZAÇÃO	7
1.2	OBJETIVO	7
1.3	JUSTIFICATIVA	7
1.4	ESTRUTURA DO TRABALHO	8
<b>2</b>	<b>DADOS GERAIS DA MONITORIA</b>	<b>9</b>
<b>3</b>	<b>CONTEÚDOS ABORDADOS</b>	<b>10</b>
3.1	LÓGICA CLÁSSICA	10
3.1.1	Proposição	10
3.1.2	Operações Lógicas	11
3.1.3	Tabelas-Verdade	13
3.1.4	Tautologia, Contradição e Contingência	15
3.1.5	Implicação Lógica	16
3.1.6	Equivalência Lógica	17
3.1.7	Álgebra das Proposições	18
3.1.8	Método Dedutivo	19
3.2	CONCEITOS SOBRE ALGOTIMOS	20
3.2.1	Conceitos Fundamentais	20
3.2.2	Tipos Primitivos de Dados	22
3.2.3	Operadores: Aritméticos, lógicos e Relacionais	22
3.2.4	Comandos de Atribuição, Entrada e Saída de Dados	24
3.2.5	Estruturas Condicionais	24
3.2.6	Estrutura de Repetição	25
<b>4</b>	<b>METODOLOGIA</b>	<b>27</b>
4.1	FERRAMENTAS DE EXECUÇÃO	27
4.1.1	Portugol Studio	27
4.1.2	Google Classroom	27
4.2	REUNIÕES DA MONITORIA	27



**INSTITUTO FEDERAL**

Rio Grande do Norte

Campus Lajes

4.2.1	Reuniões de Orientação	27
4.2.2	Reuniões com Alunato	28
4.3	RESULTADOS OBTIDOS	28
<b>5</b>	<b>ANÁLISE E DISCUSSÃO DOS RESULTADOS</b>	<b>29</b>
5.1	AQUISIÇÃO DE CONHECIMENTOS E EXPERIÊNCIAS	29
5.2	AUSÊNCIA DO ALUNATO AO LONGO DA MONITORIA	29
<b>6</b>	<b>CARACTERIZAÇÃO DAS ATIVIDADES DESENVOLVIDAS</b>	<b>30</b>
6.1	DISCIPLINA OBJETIVO	30
6.2	DESCRIÇÃO DAS FERRAMENTAS UTILIZADAS	30
<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>32</b>
	<b>REFERÊNCIAS</b>	<b>33</b>
	<b>ANEXO A – FORMULÁRIO DE IDENTIFICAÇÃO</b>	<b>34</b>

## **1 INTRODUÇÃO**

### **1.1 CONTEXTUALIZAÇÃO**

A disciplina de Fundamentos de Lógica e Algoritmo é um importante pilar para o aprendizado de programação. Quando iniciada uma nova disciplina, é normal que o aluno sofra com algum grau de dificuldade no seu aprendizado. Esse fato é evidente quando se refere a disciplina de FLA, tendo em vista que a maioria dos alunos nunca teve contato com o tema antes. O mais natural é que os alunos tenham suas dúvidas sanadas no ambiente de sala de aula através do professor, porém muitos alunos não se sentem confortáveis para fazerem perguntas em sala de aula ou mesmo não conseguem se comunicar com o professor. Outro grupo ainda em paralelo ao já citado são de alunos que à primeira vista aprendem o conteúdo, mas apresentam o surgimento de dúvidas fora do ambiente de sala, geralmente em casa durante um momento de prática ou revisão do dado conteúdo.

### **1.2 OBJETIVO**

Deste contexto surge a necessidade de um monitor. Um aluno que já possui experiência e perícia com os conteúdos da disciplina e ferramentas utilizadas, e que está disponível para tirar dúvidas fora do horário de aula em um ambiente mais à vontade. Com o objetivo de ajudar o professor, e realizar a atividade de prática profissional obrigatória ao final do curso técnico de informática.

### **1.3 JUSTIFICATIVA**

É válido ressaltar, que a prática da monitoria não se resume apenas em oferecer o suporte necessário aos discentes. A prática da monitoria propicia aos voluntários realizar atividades diretamente relacionadas com o ambiente acadêmico, proporcionando enriquecimento da formação e do currículo profissional do aluno.

#### 1.4 ESTRUTURA DO TRABALHO

Este trabalho será apresentado em capítulos, para facilitar o entendimento. O primeiro capítulo abordará a Introdução; no segundo, os Dados Gerais da Monitoria; no terceiro, os Conteúdos Abordados; no quarto, a Metodologia Utilizada; no quinto, a Caracterização das Atividades Desenvolvidas; no sexto, a Análise e Discussão dos Resultados; e por fim, as Considerações Finais.

## 2 DADOS GERAIS DA MONITORIA

Título do projeto: Monitoria da disciplina de Fundamentos de Lógica e Algoritmos.

Período de realização: 27/06/2022 a 30/12/2022.

Total de horas: 340 horas.

Orientador: Profº Me.Dannilo Martins Cunha.

Função: Monitor.

Aqui serão apresentadas de forma sucinta as **atividades previstas** ao longo do período de realização da monitoria da disciplina de FLA. Essas informações constam no plano de atividades da monitoria em questão.

Tabela 1 – Síntese das atividades do aluno na monitoria.

<b>CARGA HORÁRIA</b>	<b>ATIVIDADES DESENVOLVIDAS</b>	<b>RESULTADOS ALCANÇADOS</b>
170	Sanar algumas dúvidas pontuais dos alunos da disciplina de Fundamentos de Lógica e Algoritmos.	Estudantes Ausentes.
170	Fornecer suporte na resolução de desafios aos alunos da disciplina de Fundamentos de Logica e Algoritmos.	Estudantes Ausentes.

Fonte: Autoria própria (2023)



### **3 CONTEÚDOS ABORDADOS**

Neste capítulo serão apresentados os assuntos abordados na disciplina de FLA. Tais conteúdos podem ser divididos em dois tópicos principais, sendo o primeiro a Lógica Clássica Matemática e o segundo sendo Conceitos Iniciais Sobre Algoritmos.

#### **3.1 LÓGICA CLÁSSICA**

##### **3.1.1 Proposição**

Proposições são sentenças simples ou declarações que podem ser avaliadas como verdadeiras ou falsas. Elas são usadas na lógica matemática para formar argumentos lógicos, como a dedução, a indução e a abdução. Elas são usadas para estabelecer relações entre fatos e para avaliar, de forma lógica, a validade de uma conclusão. Elas também podem ser usadas para a solução de problemas, pois permitem ao usuário identificar as premissas necessárias para uma conclusão e estabelecer relações entre as proposições.

É importante lembrar que uma proposição não pode ser afirmada como verdadeira e falsa ao mesmo tempo, pois isso violaria a lógica matemática. Esse princípio é conhecido como “o princípio da não contradição.” Também é válido lembrar que existem dois tipos de estrutura de proposições, as proposições simples e as proposições compostas, e que ambas podem ser representadas através de letras proposicionais.

As Letras proposicionais são letras usadas para representar proposições lógicas, como verdadeiro ou falso. Por exemplo, a letra "p" poderia representar a proposição "o céu é azul". As proposições simples são representadas por letras minúsculas, enquanto proposições compostas são representadas por letras maiúsculas.

. As proposições simples são sentenças únicas e indivisíveis que podem ser avaliadas como verdadeiras ou falsas. Por exemplo: "O céu é azul."

Figura 1 – Proposições simples.

p: O céu é azul.  
q: Maçãs são doces.  
r: O número 2 é par.  
s: 4 é a raiz quadrada de 16.

Fonte: Autoria própria (2023)

Uma proposição composta é uma sentença formada por duas ou mais proposições simples conectadas. Para que possua coesão, é necessário o uso de conectivos que interliguem as proposições simples entre si, sendo os mais comuns na lógica matemática clássica, "e", "ou", "não", "se... então..." e "...se e somente se...". Por exemplo: "O céu é azul, mas as nuvens são brancas."

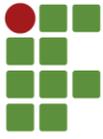
Figura 2 – Proposições compostas.

P: O céu é azul **e** as nuvens são brancas.  
Q: Maçãs são doces **ou** limões são azedos.  
R: Se Pedro é alto, **então** Joana é loira.

Fonte: Autoria própria (2023)

### 3.1.2 Operações Lógicas

As "operações lógicas" são operações semelhantes às operações matemáticas que são criadas a partir de proposições e utilizam variáveis lógicas para chegar a um resultado de verdadeiro (**V**) ou falso (**F**). Uma variável lógica é uma variável binária que pode ter apenas dois valores possíveis: verdadeiro ou falso. Uma proposição é uma declaração que pode ser considerada verdadeira ou falsa. Enquanto a variável lógica é um símbolo que representa a verdade ou falsidade de uma declaração, a proposição é a própria declaração. A variável lógica é uma abstração da proposição.



Existem cinco tipos de operações lógicas, cada qual com sua própria representação simbólica, sendo elas:

**Negação:** A negação é usada para inverter o estado de uma variável lógica. Por exemplo, se uma variável é verdadeira, a negação tornará a variável falsa. A negação é representada pelo sinal (" $\sim$ ") antes da variável.

Por exemplo, a negação de "s", que seria (não "s"), é representada como " $\sim$ s", por exemplo:

- s: o céu é azul. (**V**)
- $\sim$ s: o céu **não** é azul (**F**)

**Conjunção:** A conjunção é usada para combinar duas variáveis lógicas. Caso as duas variáveis sejam verdadeiras, a conjunção resultará em uma variável de valor verdadeiro, qualquer outro caso diferente resultara em um valor falso. A conjunção é representada por um sinal de conjunção (" $\wedge$ ") entre as variáveis, ou o conectivo "e".

Por exemplo, a conjunção de "s" e "q" é representada como  $s \wedge q$ .

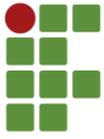
- $s \wedge q$ : O fogo é quente **e** o mar é salgado (**V**)
- $r \wedge p$ : O fogo é frio **e** o mar é salgado. (**F**)

**Disjunção:** A disjunção também é usada para combinar duas variáveis lógicas, mas de uma forma diferente da conjunção. Se uma ou outra variável for verdadeira, a disjunção resultará em valor verdadeiro, a disjunção resultara como valor falso somente se ambas as variáveis forem falsas. A disjunção é representada por um sinal de disjunção (" $\vee$ ") entre as variáveis, ou através do conectivo "ou".

Por exemplo, a disjunção de "s" e "r" é representada como  $s \vee r$ .

- $s \vee t$ : O fogo é frio **ou** o mar é salgado (**V**)
- $r \vee p$ : O fogo é quente **ou** o mar é doce (**V**)
- $s \vee p$ : O fogo é frio **ou** o mar é doce (**F**)

**Operação Condicional:** Este é um caso muito específico, onde a primeira



proposição deve ter valor verdadeiro e a segunda valor falso para resultar em uma variável lógica de valor falso, Todos os demais casos serão verdadeiros. Esta operação é indicada pelo símbolo " $\rightarrow$ ", entre duas proposições e pode ser representada pelos conectivos "se... então...".

Por exemplo, a condicional de "s" e "q" é representada como  $s \rightarrow q$ .

- $s \rightarrow q$ : Se o mar é azul então borboletas não voam (**F**)
- $s \rightarrow q$ : Se o mar é vermelho então borboletas não voam (**V**)
- $s \rightarrow q$ : Se o mar é azul então borboletas voam (**V**)
- $s \rightarrow q$ : Se o mar é rosa então borboletas voam (**V**)

Operação Bicondicional: Na operação bicondicional temos uma operação que envolve duas proposições "s" e "t". O resultado verdadeiro é obtido quando ambos os valores são verdadeiros ou ambos são falsos, caso contrário o resultado é falso. Isso é representado pelo conectivo "se e somente se" e pela notação " $s \leftrightarrow t$ ".

- $s \leftrightarrow q$ : A semana tem 8 dias **se e somente se** o céu for amarelo (**V**)
- $r \leftrightarrow t$ : A semana tem 7 dias **se e somente se** o céu for azul (**V**)
- $r \leftrightarrow q$ : A semana tem 7 dias **se e somente se** o céu for vermelho (**F**)

### 3.1.3 Tabelas-Verdade

As tabelas verdade são uma ferramenta importante na lógica de programação, pois permitem avaliar o resultado de uma expressão lógica, determinar a validade de uma proposição e identificar erros em algoritmos. Elas mostram todas as possíveis combinações de valores das variáveis booleanas envolvidas na expressão e o resultado correspondente, que pode ser verdadeiro ou falso.

Com as tabelas verdade, também é possível aplicar o princípio do terceiro excluído, que afirma que uma proposição só pode ser verdadeira ou falsa, não havendo uma terceira opção. Isso é fundamental para garantir que o comportamento de um algoritmo seja determinado de forma precisa em todas as possíveis combinações de entradas, garantindo que o programa execute corretamente em todas

as situações possíveis.

As tabelas verdade são úteis para testar a correção de expressões lógicas complexas, determinar os valores das variáveis envolvidas em um determinado ponto do programa e simplificar expressões lógicas complexas, identificando padrões em suas colunas de resultados.

Tabela 2 – Exemplo de tabela-verdade.

A	B	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
V	V	V	V	V	V
V	F	F	V	F	F
F	V	F	V	V	F
F	F	F	F	V	V

Fonte: Autoria própria (2023)

O método para determinar o número de linhas da tabela verdade a ser criada é  $2^n$ , sendo “n” o número de proposições simples que compõem a proposição composta em questão. Basta então fazer o cálculo simples.

Por exemplo:  $R:(s \wedge q)$ : O fogo é quente e o mar é salgado. A proposição composta “R” é formada por somente duas proposições simples “s” e “q”.

Figura 3 – Exemplificação do método de determinar linhas.

$2^2 = 4$ linhas
------------------

Tabela 3 – Exemplificação da proposição  $s \wedge q$ .

A	B	$A \wedge B$
V	V	V
V	F	F
F	V	F
F	F	F

Fonte: Autoria própria (2023)

Vale lembrar que os elementos da tabela-verdade são criados da direita para esquerda.

Tabela 4 – Tabela verdade da proposição  $(p \wedge q) \vee r$ .

p	q	r	$(p \wedge q) \vee r$
V	V	V	V
V	V	F	V
V	F	V	V
V	F	F	F
F	V	V	V
F	V	F	F
F	F	V	V
F	F	F	F

Fonte: Autoria própria (2023)

### 3.1.4 Tautologia, Contradição e Contingência

Existem alguns tipos especiais de proposições que precisam ser destacadas. Estas são as tautologias, as contradições e as contingências.

Uma tautologia é uma proposição que é sempre verdadeira, independentemente das circunstâncias ou do contexto em que é considerada. Por exemplo, a proposição "todos os solteiros são não casados" é uma tautologia, já que a definição de solteiro é precisamente "não casado". Todas as proposições que têm o mesmo valor lógico de uma tautologia são chamadas de "logicamente verdadeiras".

Tabela 5 – Tabela verdade de uma tautologia.

p	q	$(q \rightarrow p)$	$p \rightarrow (q \rightarrow p)$
V	V	V	V
V	F	V	V
F	V	F	V
F	F	V	V

Fonte: Matem4tica, Felipe, 2018.

Uma contradição é o oposto de uma tautologia: é uma proposição que é sempre falsa, independentemente do contexto. Por exemplo, a proposição "este círculo é quadrado" é uma contradição, já que um círculo, por definição, não pode ser quadrado. Todas as proposições que têm o mesmo valor lógico de uma contradição são chamadas de "logicamente falsas".

Tabela 6 – Tabela verdade de uma contradição.

$p$	$\sim p$	$p \wedge \sim p$
V	F	F
F	V	F

Fonte: FILHO, 1975.

Uma proposição contingente é aquela que pode ser verdadeira ou falsa, dependendo das circunstâncias ou do contexto. Por exemplo, a proposição "hoje está chovendo" é contingente, pois pode ser verdadeira em alguns lugares e falsa em outros. Todas as proposições que não são nem tautologias nem contradições são chamadas de "contingentes".

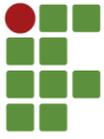
Tabela 7 – Tabela verdade de uma contingência (Negação)

$p$	$q$	$\sim q$	$p \wedge \sim q$
V	V	F	F
V	F	V	V
F	V	F	F
F	F	V	F

Fonte: Matem4tica, Felipe, 2018.

### 3.1.5 Implicação Lógica

A implicação lógica pode ser definida como uma relação entre duas proposições, onde a primeira proposição é chamada de antecedente e a segunda proposição é chamada de conseqüente. A implicação lógica afirma que se o antecedente é verdadeiro, então o conseqüente também deve ser verdadeiro. Se o



antecedente é falso, então a implicação não impõe nenhuma restrição sobre o valor de verdade do consequente.

Por exemplo, considere a implicação lógica "se está chovendo, então as ruas estão molhadas". Neste caso, a proposição "está chovendo" é o antecedente e a proposição "as ruas estão molhadas" é o consequente. A implicação lógica afirma que se está chovendo, então as ruas estão molhadas. Se não está chovendo, então não podemos inferir nada sobre se as ruas estão molhadas ou não.

Além das propriedades, a implicação lógica também possui regras de inferência, que delimitam as operações usadas para encontrar proposições dentro de outras proposições. Existem múltiplas regras de inferência, mas é válido citar algumas importantes, tais como:

- Adição ( $p \Rightarrow p \vee q$  e  $q \Rightarrow p \vee q$ );
- Simplificação ( $p \wedge q \Rightarrow p$  e  $p \wedge q \Rightarrow q$ );
- Silogismo Disjuntivo ( $(p \vee q) \wedge \sim q \Rightarrow p$ );
- Modus Ponens ( $(p \rightarrow q) \wedge p \Rightarrow q$ );
- Modus Tollens ( $(p \rightarrow q) \wedge \sim q \Rightarrow \sim p$ );

### 3.1.6 Equivalência Lógica

Uma equivalência lógica se dá quando duas distintas proposições compostas possuem tabelas-verdade idênticas. As proposições "Se está chovendo, as ruas estão molhadas" e "As ruas estão molhadas se e somente se está chovendo" são equivalentes, pois elas têm o mesmo valor lógico em todas as situações possíveis. Se está chovendo, então as ruas estão molhadas e se as ruas estão molhadas, então está chovendo. Se não está chovendo e as ruas estão secas, então ambas as proposições são falsas.

- Reflexividade: Uma proposição é sempre equivalente a si mesma. Isso pode ser representado pela seguinte propriedade:  $A \leftrightarrow A$ .
- Simetria: Se A é equivalente a B, então B é equivalente a A. Isso pode ser



representado pela seguinte propriedade:  $A \leftrightarrow B$  implica  $B \leftrightarrow A$ .

- Transitividade: Se  $A$  é equivalente a  $B$  e  $B$  é equivalente a  $C$ , então  $A$  é equivalente a  $C$ . Isso pode ser representado pela seguinte propriedade:  
 $A \leftrightarrow B$  e  $B \leftrightarrow C$  implicam  $A \leftrightarrow C$ .

Além de propriedades, existem algumas regras para a equivalência lógica. Entre elas podemos citar três:

- Dupla Negação ( $p \leftrightarrow \sim\sim p$ );
- Regra de CLAVIUS ( $\sim p \rightarrow p \leftrightarrow p$ );
- Regra da Absorção ( $p \rightarrow p \wedge q \leftrightarrow p \rightarrow q$ );

### 3.1.7 Álgebra das Proposições

A chamada Álgebra das Proposições incorpora diferentes propriedades matemáticas aos diversos tipos de proposições. (FILHO, 1975) - Iniciação à Lógica Matemática - Na obra, são apresentadas as propriedades da conjunção, da disjunção e da conjunção e disjunção combinadas, além das negações da condicional e da bicondicional.

Na conjunção temos as propriedades idempotente ( $p \wedge p \leftrightarrow p$ ), comutativa ( $p \wedge q \leftrightarrow q \wedge p$ ), associativa [ $(p \wedge q) \wedge r \leftrightarrow p \wedge (q \wedge r)$ ], identidade (“ $p \wedge t \leftrightarrow p$ ” e “ $p \wedge c \leftrightarrow c$ ”), onde “ $t$ ” sempre é verdadeiro (elemento neutro) e “ $c$ ” sempre é falso (elemento absorvente).

Na disjunção temos as propriedades idempotente ( $p \vee p \leftrightarrow p$ ), comutativa ( $p \vee q \leftrightarrow q \vee p$ ), associativa [ $(p \vee q) \vee r \leftrightarrow p \vee (q \vee r)$ ], identidade (“ $p \vee t \leftrightarrow t$ ” e “ $p \vee c \leftrightarrow p$ ”), onde “ $t$ ” sempre é verdadeiro (elemento absorvente) e “ $c$ ” sempre é falso (elemento neutro).

Também existem propriedades para proposições que possuem tanto conjunções quanto disjunções. As primeiras são as Distributivas, onde é definido que a tabela-verdade de “ $p \wedge (q \vee r)$ ” é igual à de “ $(p \wedge q) \vee (p \wedge r)$ ” e também que a proposição “ $p \vee (q \wedge r)$ ” é equivalente à “ $(p \vee q) \wedge (p \vee r)$ ”; a segunda é a Absorção, onde “ $p \wedge (p \vee q) \leftrightarrow p$ ” e “ $p \vee (p \wedge q) \leftrightarrow p$ ”; e por fim as Regras de Morgan, que se subdivide em duas:



- 1) Negar que duas proposições são verdadeiras ao mesmo tempo equivale a afirmar que pelo menos uma delas é falsa.
- 2) Negar que pelo menos uma dentre duas proposições é verdadeira equivale a afirmar que ambas são falsas. (FILHO, 1975, p. 73)

Ainda existem duas propriedades com negação, são elas: negação da condicional, onde a proposição " $\sim(p \rightarrow q)$ " é equivalente a " $p \wedge \sim q$ "; e negação da bicondicional, onde a proposição " $\sim(p \leftrightarrow q)$ " é equivalente a " $(p \wedge \sim q) \vee (\sim p \wedge q)$ ".

### 3.1.8 Método Dedutivo

Embora seja comum representar uma implicação ou equivalência lógica através do "Método das tabelas-verdade", existe outra maneira eficiente de demonstrar essas equivalências, chamada de "Método dedutivo".

Diferentemente do método das tabelas-verdade, que exige a construção de uma tabela para cada expressão lógica, o "Método dedutivo" é um método mais prático que permite a redução de conectivos usados em uma expressão, tal redução é realizada seguindo os princípios das leis de Morgan.

A primeira lei de Morgan diz que negar uma conjunção é o mesmo que a transformar em uma disjunção. Exemplo:  $\sim(p \wedge q) = (\sim p) \vee (\sim q)$

A segunda lei de Morgan diz que negar uma disjunção é o mesmo que a transformar em uma conjunção. Exemplo:  $\sim(p \vee q) = (\sim p) \wedge (\sim q)$

Essa redução é útil na busca pela forma normal de uma proposição, que é quando a proposição contém apenas os conectivos " $\sim$ ", " $\wedge$ ", e " $\vee$ " em sua estrutura. A forma normal pode ser dividida em duas subcategorias: a Forma Normal Conjuntiva (FNC) e a Forma Normal Disjuntiva (FND).

Figura 4 – Redução de conectivos (Simplificação de implicação)

Proposição composta: $(p \vee q) \rightarrow r$ FNC: $\sim (p \vee q) \vee r$ FND: $(\sim p \vee r) \wedge (\sim q \vee r)$
---

Fonte: A autoria própria (2023)

## 3.2 CONCEITOS SOBRE ALGORITMOS

### 3.2.1 Conceitos Fundamentais

Diferentes autores oferecem definições distintas para o que é um algoritmo. No entanto, de maneira geral, um algoritmo pode ser descrito como uma sequência finita de instruções lógicas que devem ser seguidas para atingir um objetivo predefinido.

Em seu livro, (ASCENCIO & CAMPOS) apresentam alguns exemplos simples de algoritmos, incluindo três tipos diferentes. O primeiro deles é a descrição narrativa, representada na Figura 4, que, como o nome sugere, descreve as etapas de um processo de forma detalhada utilizando linguagem natural. Embora seja uma forma simples de representar um algoritmo, pode ser difícil de ser seguida ou interpretada por outras pessoas.

Figura 5 – Algoritmo para multiplicar dois valores (Descrição Narrativa)

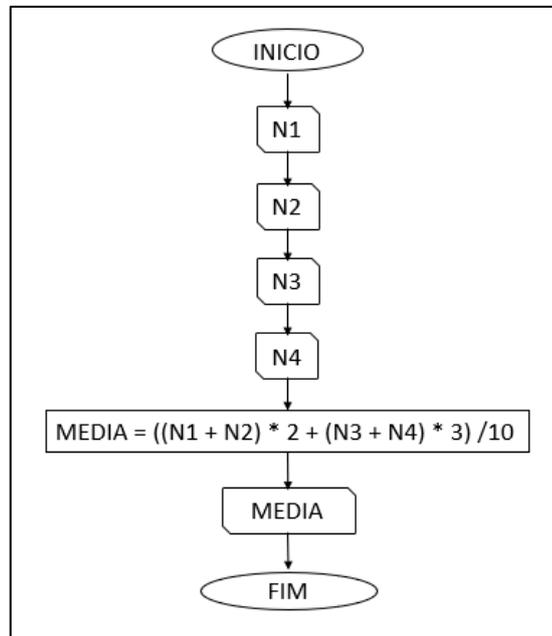
- |  |
|--|
| <ol style="list-style-type: none"><li>1. Receba dois números</li><li>2. Multiplique o primeiro valor pelo segundo</li><li>3. Escreva o resultado da multiplicação.</li></ol> |
|--|

Fonte: A autoria própria (2023)

O segundo tipo é o Fluxograma, que utiliza diagramas gráficos para representar as etapas de um processo. Cada etapa é representada por um símbolo específico, e as linhas conectam os símbolos para indicar a ordem das etapas, como mostrado na Figura 5. É uma forma visual de representar um algoritmo que pode ser facilmente

compreendida por outras pessoas.

Figura 6 – Algoritmo para descobrir a média final ponderada (Fluxograma).



Fonte: Autoria própria (2023)

O terceiro e último tipo é chamado de pseudocódigo, que faz uso de regras preexistentes para representar o passo a passo para a solução do problema trabalhado, como exemplificado na Figura 6, onde o código foi feito no Portugol Studio.

Figura 7 – Algoritmo para saber a média final comum (pseudocódigo)

```
programa
{
    funcao inicio()
    {
        inteiro nota1, nota2, nota3, nota4, soma

        escreva("Informe a primeira nota:")
        leia(nota1)

        escreva("Informe a primeira nota:")
        leia(nota2)

        escreva("Informe a primeira nota:")
        leia(nota3)

        escreva("Informe a primeira nota:")
        leia(nota4)

        soma = (nota1 + nota2 + nota3 + nota4)/4
        // soma todas as notas e divide por quatro

        escreva("A sua media final comum é:" + soma)
        // exibe o resultado da soma
    }
}
```



Fonte: Autoria própria (2023)

### **3.2.2 Tipos Primitivos de Dados**

Na programação, há uma grande variedade de classificações de dados, mas existem três tipos primitivos que estão presentes em praticamente todos os algoritmos: os numéricos, lógicos e literais.

Os dados numéricos, por sua vez, são divididos em dois subtipos: inteiros e reais. Os inteiros podem ser considerados como variáveis que armazenam números inteiros, ou seja, valores positivos, negativos ou nulos que não possuem casas decimais. Já os reais armazenam números reais na forma de frações decimais, podendo ser positivos ou negativos, e são separados por pontos. Ao contrário dos inteiros, os dados reais possuem casa decimal.

Os dados lógicos, também conhecidos como booleanos, são os mais simples dos tipos primitivos. As variáveis lógicas podem assumir apenas dois valores: verdadeiro ou falso, assim como as proposições lógicas.

Por fim, os dados literais podem ser subclassificações em dois tipos: caractere e cadeia de caracteres. Os dados do tipo caractere armazenam um único caractere, enquanto os do tipo cadeia armazenam um conjunto de caracteres que podem formar palavras ou frases. Ambos podem armazenar letras maiúsculas, letras minúsculas, números e caracteres especiais, mas não podem ser utilizados em cálculos.

### **3.2.3 Operadores: Aritméticos, lógicos e Relacionais**

Os operadores são elementos fundamentais e indispensáveis na construção de algoritmos. Eles se dividem em três tipos principais: aritméticos, lógicos e relacionais.

Os operadores aritméticos são aqueles utilizados em expressões matemáticas, ou seja, cálculos matemáticos. Os operadores estão sendo representados na Tabela 8 com seus respectivos valores de prioridade. Lembrando que somente os operadores básicos são utilizados em cálculos matemáticos, sendo assim “++” e “--” são operadores utilizados somente na programação.

Tabela 8 – Operadores aritméticos.

Operação	Símbolo	Prioridade
Adição	+	1
Subtração	-	1
Multiplicação	*	2
Divisão	/	2
Resto da divisão	%	2
Incremento	++	2
Decremento	--	2

Fonte: Autoria própria (2023)

Os operadores lógicos são usados em expressões lógicas, ou seja, proposições ou variáveis do tipo lógico. Dessa forma, os valores associados aos operadores precisam ser verdadeiros ou falsos. Segue a Tabela 9 aonde estão representados os operadores, seus respectivos símbolos utilizados na linguagem de programação e seus valores de prioridade.

Tabela 9 – Operadores lógicos.

Operador	Símbolo	Prioridade
ou		1
e	&&	2
não	!	3

Fonte: Autoria própria (2023)

Os operadores relacionais são utilizados quando se quer comparar os valores de variáveis ou outras expressões em um programa. Eles retornam um valor lógico, que pode ser verdadeiro ou falso, dependendo do resultado da comparação.

Eles são comumente usados com valores numéricos, mas também podem ser usados com outros tipos de dados, como caracteres ou strings, que são cadeias de caracteres. A Tabela 10 mostra os operadores em questão.



Tabela 10 – Operadores relacionais.

Operação	Símbolo
Maior	>
Menor	<
Maior igual	>=
Menor igual	<=
igual	==
Diferente	!=

Fonte: Autoria Própria (2023)

Os operadores possuem uma hierarquia de ordem para ser seguida, tanto dentro do próprio tipo quanto entre os tipos de operadores. Sempre são resolvidos primeiro os aritméticos (seguindo a ordem de prioridade da Tabela 8), depois os relacionais e por último os lógicos (seguindo a ordem de prioridade da Tabela 9)

### 3.2.4 Comandos de Atribuição, Entrada e Saída de Dados

Escrever um algoritmo é uma tarefa que exige a criação e utilização de variáveis para o armazenamento de dados. Para salvar qualquer tipo de conteúdo em uma variável é necessário a utilização de um comando de entrada, dentre tais o mais comum utilizado é o "=", já que todos os outros comandos são derivados do mesmo.

Existem também os comandos de entrada e saída de dados, que são indispensáveis. O comando de entrada é utilizado para solicitar dados ao usuário, que após recebidos serão armazenados em variáveis. No Portugol este comando representado como "leia".

O comando de saída é utilizado para mostrar informações ao usuário, que podem ser mensagens ou dados produzidos pelo algoritmo. Este comando é representado no Portugol como "escreva".

### 3.2.5 Estruturas Condicionais

As estruturas condicionais são utilizadas pelos programadores para estabelecer condições que determinam a execução de uma sequência específica de

códigos. Se a condição especificada for verdadeira, um bloco de comandos é executado. Caso contrário, os comandos condicionais são ignorados.

Essas condições são definidas como expressões e podem ser formadas utilizando operadores aritméticos, lógicos e, principalmente, relacionais. Além de números e variáveis, esses operadores permitem que o programador crie condições mais complexas que envolvem múltiplas variáveis e valores.

Figura 8 – Algoritmo de verificação de idade no Portugol (pseudocódigo)

```
1 programa
2 {
3
4   funcao inicio()
5   {
6       inteiro idade
7
8       escreva("Digite sua idade: ")
9       leia(idade)
10
11       se(idade > 18) {
12           escreva("Você é maior de idade.")
13       }
14       senao {
15           escreva("Você é menor de idade.")
16       }
17   }
18 }
```

Fonte: Autoria própria (2023)

### 3.2.6 Estrutura de Repetição

Em muitas situações de programação, é necessário repetir uma sequência de códigos diversas vezes. Quando essa repetição ocorre muitas vezes ou quando o número de repetições pode variar de acordo com o usuário, é utilizado uma estrutura de repetição.

Assim como as estruturas condicionais, as estruturas de repetição exigem uma condição para determinar se a sequência de códigos deve continuar sendo repetida. Essa condição é expressa por meio de uma expressão lógica. Enquanto essa expressão for avaliada como verdadeira, a repetição continuará.

Figura 9 – Algoritmo para montar tabuada no Portugol (Pseudocódigo)

```
1 programa
2 {
3
4     funcao inicio()
5     {
6         inteiro numero, resultado, contador
7
8         escreva("Informe um número para ver sua tabuada: ")
9         leia(numero)
10
11        limpa()
12
13        para (contador = 1; contador <=10; contador ++)
14        {
15            resultado = numero * contador
16            escreva (numero, "X", contador, "=", resultado, "\n" )
17        }
18    }
19 }
```

Fonte: Biblioteca Portugol, UNIVALI (2014)



## **4 METODOLOGIA**

Este capítulo se destina a descrever as etapas e os procedimentos metodológicos utilizados para a realização desta monitoria, caracterizando-a e apontando os instrumentos utilizados na sua prática.

### **4.1 FERRAMENTAS DE EXECUÇÃO**

#### **4.1.1 Portugol Studio**

O aplicativo Portugol Studio é a ferramenta que deveria ser utilizada como forma de prática para o desenvolvimento dos alunos, através de exemplos e desafios simples que acompanhariam os ensinamentos passados ao longo das aulas ministradas, além de servir para sanar as dúvidas dos alunos.

#### **4.1.2 Google Classroom**

Foi criada uma sala de aula dentro da plataforma Google Classroom com fins de facilitar o contato entre os monitores e o professor orientador. Nela foram disponibilizadas, em forma de PDF, referências para estudo de lógica e o modelo dos relatórios mensais.

### **4.2 REUNIÕES DA MONITORIA**

#### **4.2.1 Reuniões de Orientação**

As reuniões foram realizadas tanto de forma presencial como online. As reuniões presenciais geralmente eram realizadas às oito horas da manhã, enquanto as reuniões online eram realizadas às dez horas, através da plataforma Google Meet.

As reuniões online serviram para explicar aos monitores sobre o seu papel de

sanar dúvidas dos alunos, os documentos que seriam necessários e as ferramentas para se comunicar com o professor e entre si. As reuniões presenciais serviram para maior esclarecimento a respeito da monitoria no geral, em que local do campus ela seria realizada, os horários de atuação dos monitores, como seriam os relatórios mensais e a postura dos monitores a respeito das dúvidas dos alunos.

#### **4.2.2 Reuniões com Alunato**

As reuniões eram realizadas de segunda-feira à sexta-feira sempre no mesmo horário, 8:30h às 11:30. Todas as reuniões foram realizadas de forma presencial no ambiente da biblioteca do campus.

Nesses encontros os monitores aguardavam a chegada dos discentes, que trariam suas dúvidas quanto às atividades ou conteúdos passados na disciplina de FLA.

### **4.3 RESULTADOS OBTIDOS**

A monitoria serviu de oportunidade para pôr em prática e exercitar os conhecimentos adquiridos através das aulas da disciplina de FLA ao longo do curso. Ela permitiu a aquisição de novos conhecimentos e experiências, tanto da parte dos monitores quanto dos alunos.

Apesar dos esforços dos monitores e do professor orientador o resultado colhido através da monitoria foi negativo, pois todos alunos se mantiveram ausentes durante todo o período.

## **5 ANÁLISE E DISCUSSÃO DOS RESULTADOS**

Neste capítulo serão analisados e discutidos, de forma mais aprofundada, os resultados obtidos através da realização desta monitoria voluntária.

### **5.1 AQUISIÇÃO DE CONHECIMENTOS E EXPERIÊNCIAS**

Com o decorrer da monitoria foi possível aprender mais acerca da disciplina, além de obter novas experiências acerca da gestão organizacional, bem como a área da docência. Ler materiais e realizar pesquisas para compreender os conteúdos abordados pela matéria, além de seguir horários pré-estabelecidos para disponibilidade aos alunos foram costumes necessários obtidos para a realização desta monitoria.

Ademais, trocar conhecimentos e experiências com outros monitores da matéria de FLA, são alguns dos bons resultados obtidos com essa prática, que permitiu exercitar o que foi aprendido durante a formação acadêmica do monitor

### **5.2 AUSÊNCIA DO ALUNATO AO LONGO DA MONITORIA**

Ao longo da execução das atividades da monitoria, foi observado um grande problema no desenvolvimento da prática profissional em questão: a ausência dos discentes. Embora o professor tenha apresentado os monitores da disciplina FLA em sala de aula e exortado os alunos a procurá-los para sanar suas dúvidas, nenhum aluno compareceu à monitoria ao longo de todo o período.



## 6 CARACTERIZAÇÃO DAS ATIVIDADES DESENVOLVIDAS

Neste capítulo será apresentado, de forma detalhada, as atividades executadas durante o período de execução da monitoria.

### 6.1 DISCIPLINA OBJETIVO

A disciplina objetivo desta monitoria é a disciplina de Fundamentos de Lógica e Algoritmos, uma das matérias necessárias para se tornar técnico em informática. Com uma carga horária de 75 horas-relógio, correspondendo a 100 horas-aula, seu objetivo é apresentar aos alunos os conceitos, métodos e técnicas que são essenciais na construção de algoritmos.

Ela tem por objetivo capacitar os discentes a compreender a lógica da programação, criar algoritmos, aprender sobre as estruturas básicas de dados, entender e aplicar os operadores lógicos da álgebra booleana, bem como interpretar problemas de lógica proposicional.

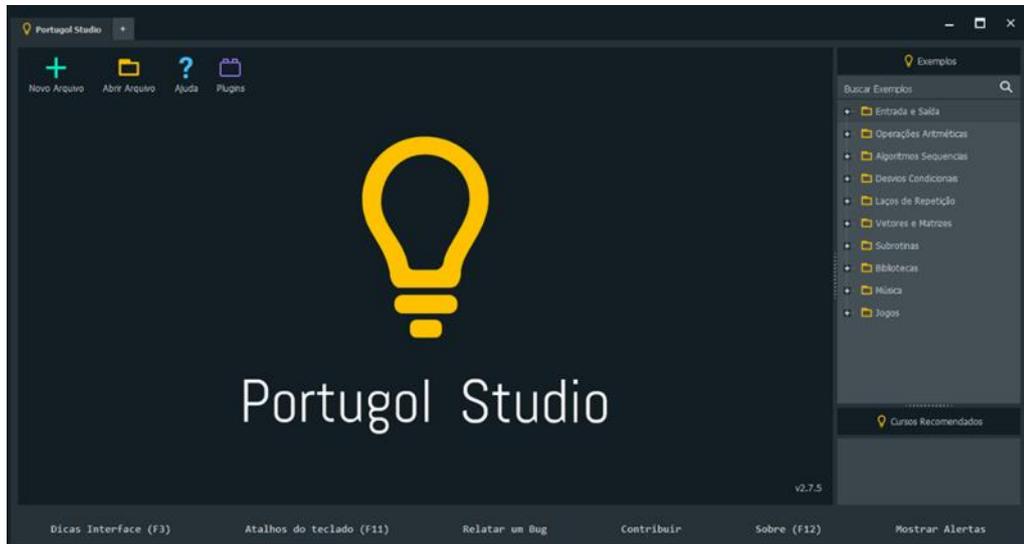
As bases científico-tecnológicas, também conhecidas como conteúdos, são:

- Introdução a Lógica Matemática;
- Implicação Lógica;
- Equivalência Lógica;
- Álgebra das Proposições;
- Método Dedutivo;
- Conceitos de Implementação de Algoritmos.

### 6.2 DESCRIÇÃO DAS FERRAMENTAS UTILIZADAS

O Portugol Studio é uma ferramenta gratuita e de código aberto que oferece recursos para criar, compilar e executar programas. É uma plataforma de pseudocódigo completamente em português, pensada para iniciantes em programação.

Figura 10 – Tela de início do Portugol Studio.



Fonte: Autoria própria (2023)

## **7 CONSIDERAÇÕES FINAIS**

O presente relatório apresentou o desempenho da monitoria de Fundamentos de Lógica e Algoritmo no Campus Avançado Lajes do IFRN. O objetivo da monitoria foi auxiliar o professor e sanar as dúvidas dos discentes, além de proporcionar a prática profissional obrigatória ao final do curso técnico em informática.

A prática profissional foi de grande importância, pois permitiu que os monitores voluntários executem atividades relacionadas com o ambiente acadêmico, enriquecendo a formação e o currículo profissional dos alunos, estimulando a cooperação entre estudantes e professores e promovendo a troca de conhecimento. Além disso, as atividades desempenhadas permitiram que os monitores aprimorassem e adquirissem novos conhecimentos acadêmicos, juntamente com novas experiências de organização, disciplina e comprometimento.

Entretanto, apesar das tentativas dos monitores e do professor em chamar a atenção dos discentes para a monitoria, foi observada uma ausência significativa dos alunos durante a prática. Em consequência disso, percebe-se que os objetivos da monitoria foram parcialmente atingidos. Porém, é importante ressaltar que tanto os docentes quanto os monitores não têm controle sobre o fluxo de requisições dos alunos e que esse resultado já era esperado. Além disso, essa experiência também agregou à formação dos profissionais envolvidos.

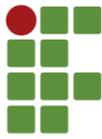


## REFERÊNCIAS

FILHO, Edgard de Alencar. Iniciação à Lógica Matemática. 2003 ed., vol. 1, São Paulo, AMPUB Comercial Ltda, 1975. 1 vols.

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi. Fundamentos da Programação de Computadores. vol. 1, São Paulo, Pearson Prentice Hall, 2016. 1 vols.

BLOGSPOT. Felipe. Raciocínio lógico: tautologias, diferenças e contingência. Matem4tica. 20 de novembro, 2018. Disponível em: <<https://matem4tica.blogspot.com/2018/11/raciocinio-logico-tautologias-contradicao.html>>. Acesso em: 27 mar. 2023.



### ANEXO A – FORMULÁRIO DE IDENTIFICAÇÃO

Dados do Relatório Científico	
Título e subtítulo: Relatório da Prática Profissional: Monitoria de Fundamentos de Lógica e Algoritmos	
Tipo de relatório: Monitoria	Data: 27/06/2022 a 30/12/2022.
Título do projeto/ programa/ plano: Monitoria de Fundamentos de Lógica e Algoritmos	
Autor(es): Luiz Felipe das Chagas Bezerra	
Instituição e endereço completo: Instituto Federal do RN Campus Lajes Endereço: BR-304, Km 120, s/n - Centro, Lajes - RN, 59535-000	
Resumo: A pratica profissional em questão foi caracterizada na forma de uma monitoria, que teve como objetivo auxiliar os iniciados no curso de Informática, tendo em vista que a disciplina de Fundamentos de Logica e Algoritmo é a base para os jovens na área da programação. As atividades realizadas serviram principalmente para o enriquecimento curricular dos munitores voluntários, uma vez que executaram atividades relacionadas ao âmbito acadêmico. A presente monitoria foi realizada de maneira presencial. Os objetivos foram em parte atingidos, uma vez que permitiu aos monitores se desenvolverem como profissionais, mas a ausência dos iniciados em informática é uma falha presente no desenvolvimento desta atividade de prática profissional. É valido ressaltar que tal ausência era algo esperado devido ao fato que os iniciados não costumam procurar os monitores, mas ainda assim a pratica agregou aos monitores responsáveis. Por fim os resultados obtidos foram o aprimoramento dos conhecimentos acadêmicos, ganho de experiência com as ferramentas em questão e comprometimento.	
Palavras-chave: Monitoria. Logica. Algoritmos. Pratica profissional.	
Nº de páginas: 35	
Jornada de trabalho: 3 Horas Diárias	Horas semanais: 15 Horas
Total de horas: 340 Horas	
Observações/notas:	