

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

NATANAEL ASAFE JALES DA COSTA

RELATÓRIO DA PRÁTICA PROFISSIONAL: MONITORIA DE FLA

LAJES/RN  
2023

NATANAEL ASAFE JALES DA COSTA

**RELATÓRIO DA PRÁTICA PROFISSIONAL: MONITORIA DE FLA**

Relatório de Prática Profissional apresentado ao Curso Técnico Subsequente 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

## RESUMO

Esta prática profissional em questão foi caracterizada na forma de uma monitoria, que teve como objetivo auxiliar os iniciados da área de informática e permitiu aos monitores voluntários executar atividades relacionadas com o ambiente acadêmico, enriquecendo a formação do currículo profissional dos mesmos. Tais objetivos foram alcançados através de reuniões regulares com o auxílio de ferramentas digitais. Percebeu-se, então, que os objetivos foram por parte atingidos, pois permitiram o desenvolvimento do monitor como profissional, apesar da falha no desenvolvimento da prática causada pela ausência dos alunos iniciados em informática. Contudo, é válido lembrar que, como não se têm controle sobre a procura dos alunos, tal resultado poderia ser esperado. Por fim, os resultados obtidos com a prática foram a aquisição e aprimoramento dos conhecimentos acadêmicos, ganho de experiências em organização, disciplina e comprometimento.

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

## LISTA DE ILUSTRAÇÕES

Figura 1 – Proposição Simples	9
Figura 2 – Proposições Compostas	10
Figura 3 – Emprego dos Conectivos	10
Figura 4 – Diagrama uma proposição composta	12
Figura 5 – Exemplos de redução dos números de conectivos	18
Figura 6 – Algoritmo para somar três números (Descrição Narrativa)	19
Figura 7 – Algoritmo para somar três números (Fluxograma)	20
Figura 8 – Algoritmo para somar três números (Pseudocódigo no Portugol)	20
Figura 9 – Algoritmo verificador de nota no Portugol (pseudocódigo)	24
Figura 10 – Algoritmo para montar tabuada no Portugol (Pseudocódigo)	25
Figura 11 – Tela de início do Portugol <i>Studio</i>	

## LISTA DE TABELAS

Tabela 1 – Síntese das atividades do aluno na monitoria	
Tabela 2 – Tabela verdade da proposição $(p \wedge q) \vee r$	15
Tabela 3 – Tabela-verdade de uma tautologia	15
Tabela 4 – Tabela-verdade de uma contradição	16
Tabela 5 – Tabela-verdade de uma contingência (Negação)	16
Tabela 6 – Tabelas-verdade de " $c \rightarrow p$ " e " $p \rightarrow t$ "	19
Tabela 7 – Operadores aritméticos básicos	24
Tabela 8 – Operadores lógicos básicos	24
Tabela 9 – Operadores relacionais básicos	25

## SUMÁRIO

1.INTRODUÇÃO	7
1.1. CONTEXTUALIZAÇÃO	7
1.2. OBJETIVO	7
1.4. ESTRUTURA DO TRABALHO	8
2.DADOS GERAIS DA MONITORIA	9
3.CONTEÚDOS ABORDADOS	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
3.1.6. Equivalência Lógica	17
3.1.7. Álgebra das Proposições	18
3.1.8. Método Dedutivo	19
3.2. CONCEITOS SOBRE ALGORITMOS	21
3.2.1. Conceitos Fundamentais	21
3.2.2. Tipos Primitivos de Dados	23
3.2.3. Operadores: Aritméticos, Lógicos e Relacionais	24
3.2.4. Comandos de Atribuição, Entrada e Saída de Dados	26
3.2.5. Estruturas Condicionais	26
3.2.6. Estruturas de Repetição	27
4.METODOLOGIA	28
4.1. FERRAMENTAS DE EXECUÇÃO	28
4.1.1. QUADRO BRANCO	28
4.1.2. MARCADOR (PILOTO)	29
4.1.3. Portugol <i>Studio</i>	29
4.2. REUNIÕES DA MONITORIA	29
4.2.1. Reuniões de Orientação	29
4.2.2. Encontro com Alunato	30

5.CARACTERIZAÇÃO DAS ATIVIDADES DA MONITORIA	30
5.1. DISCIPLINA-OBJETO	30
5.2. DESCRIÇÃO DAS FERRAMENTAS UTILIZADAS	31
5.3. RESULTADOS OBTIDOS	33
6.ANÁLISE E DISCUSSÃO DOS RESULTADOS	34
6.1. AQUISIÇÃO DE CONHECIMENTOS E EXPERIÊNCIAS	34
6.2. AUSÊNCIA DO ALUNATO AO LONGO DA MONITORIA	34
7.CONSIDERAÇÕES FINAIS	36
REFERÊNCIAS	38
ANEXO A – FORMULARIO DE IDENTIFICAÇÃO	39

## **1 INTRODUÇÃO**

### **1.1. CONTEXTUALIZAÇÃO**

Ao se iniciar uma nova disciplina, é normal que os alunos sintam alguma dificuldade em aprender o conteúdo. Esse fato se torna mais evidente ao observarmos discentes que estão começando matérias como FLA - um importante pilar para o aprendizado de programação -, tendo em vista que a maioria deles nunca teve contato com o tema. Desse modo, se faz necessário que esses estudantes busquem formas de tirarem suas dúvidas. Contudo, a maioria deles não se sente confortável ao fazer perguntas em sala de aula ou não conseguem se comunicar com o professor.

### **1.2. OBJETIVO**

Desse contexto é que surge a necessidade de um monitor: um aluno que já possui experiência e domínio da disciplina, e que está disponível para tirar dúvidas dos alunos em questão. Tendo como o seu objetivo auxiliar o professor, e realizar a atividade da prática profissional obrigatória ao final do curso técnico em informática, os monitores oferecem suporte aos novos estudantes de FLA fora do horário de aula.

### **1.3. JUSTIFICATIVA**

É válido ressaltar, que a prática da monitoria não se resume apenas em oferecer o suporte necessário aos discentes. Em suma, a prática da monitoria permite aos voluntários realizar atividades diretamente relacionadas com o ambiente acadêmico. Ademais, proporciona enriquecimento da formação e do currículo profissional do aluno, estimula a cooperação entre estudante e professor e promove a troca de conhecimento.

### **1.4. ESTRUTURA DO TRABALHO**

O presente trabalho, que será apresentado nas páginas a seguir, seguirá uma abordagem de divisão em capítulos para facilitar seu entendimento. De tal modo, teremos como primeiro capítulo a Introdução; no segundo teremos os Dados Gerais da Monitoria; no terceiro serão apresentados os Conteúdos Abordados; no quarto será explicada a Metodologia usada; no quinto terá a Caracterização das atividades Desenvolvidas; o sexto apresentará a Análise e Discussão dos resultados; e no sétimo serão dadas as Considerações Finais.

## 2 DADOS GERAIS DA PESQUISA/EXTENSÃO/ESTÁGIO

Título do projeto: Monitoria da disciplina de Fundamentos de Lógica e Algoritmos Período de realização: de 01/07/2022 a 15/02/2023

Total de horas: 360 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>
180 horas	Sanar as dúvidas dos alunos na disciplina de FLA.	Dúvidas resolvidas.
180 horas	Fornecer suporte na resolução de desafios extras aos alunos da disciplina de FLA.	Desafios resolvidos

Fonte: autoria própria (2023).

### 3. CONTEÚDOS ABORDADOS

Neste capítulo serão apresentados os assuntos abordados pela disciplina de FLA. O conteúdo desta matéria pode ser dividido em dois tópicos principais, sendo o primeiro a Lógica Clássica Matemática e o segundo sendo os Conceitos Iniciais Sobre Algoritmos.

#### 3.1. LÓGICA CLÁSSICA

##### 3.1.1. Proposição

De acordo com (FILHO, 1975), em seu livro Iniciação à Lógica Matemática, uma proposição é um agrupamento de palavras, símbolos ou números que representa uma ideia que possui sentido completo. As proposições expressam um pensamento ou fato que pode ser considerado verdadeiro ou falso. Toda premissa lógica segue o princípio do terceiro excluído, que indica que a proposição somente pode assumir o valor de verdadeiro ou falso, jamais um terceiro.

Contudo, é válido frisar que uma proposição não pode ser declarada como verdadeira e falsa ao mesmo tempo, pois isso quebraria a lógica matemática. Tal regra é conhecida como princípio da não contradição.

Além disso, as proposições recebem classificações quanto a sua estrutura. As chamadas proposições simples, por exemplo, são aquelas que não apresentam mais de uma proposição em sua composição. Elas são designadas a letras minúsculas - p, q, r, s... - chamadas de letras proposicionais, ilustradas na Figura 1.

Figura 1 - Proposições Simples. Fonte: FILHO, 1975.

p : Carlos é careca
q : Pedro é estudante
r : O número 25 é quadrado perfeito

Fonte: FILHO, 1975.

Também existem as chamadas proposições compostas, ou fórmulas proposicionais, que são construídas a partir da união de duas ou mais proposições. Estas são representadas por letras maiúsculas ao invés de minúsculas - P, Q, R, S...

- como os exemplos dados na Figura 2. Caso seja desejado, é possível mostrar que uma fórmula é formada por proposições simples específicas usando a representação:  $Q(r, s, p\dots)$ .

Figura 2 - Proposições compostas.

P : Carlos é careca e Pedro é estudante
Q : Carlos é careca ou Pedro é estudante
R : Se Carlos é careca, então é infeliz

Fonte: FILHO, 1975.

Para que uma proposição composta possua coesão ela precisará de palavras que liguem as proposições simples entre si. Esses termos são chamados de conectivos, e os mais comuns utilizados na lógica matemática clássica são “e”, “ou”, “não”, “se... então...” e “... se e somente se...”, como mostrado na Figura 3.

Figura 3 - Emprego dos Conectivos.

P : O número 6 é par e o número 8 é quadrado perfeito
Q : O triângulo ABC é retângulo ou é isósceles
R : Não está chovendo
S : Se Jorge é engenheiro, então sabe Matemática
T : O triângulo ABC é equilátero se e somente se é equiângulo

Fonte: FILHO, 1975.

### 3.1.2. Operações Lógicas

As operações feitas a partir de proposições recebem o nome de operações lógicas. Elas seguem regras similares às da aritmética, chamada de cálculo proposicional. As operações lógicas essenciais são a negação, conjunção, disjunção, condicional e bicondicional (FILHO, 1975, p. 17).

A negação de uma proposição “s”, também chamada de “não s”, apresentará um valor lógico oposto ao de “s”, ou seja, quando “s” for verdadeira (V) “não s” será falsa (F) e vice-versa. Sua forma padrão de notação é “ $\sim s$ ”, por exemplo:

- s : O céu é azul (V)
- $\sim s$  : O céu não é azul (F)

A conjunção é por sua vez é uma operação de duas proposições “s” e “t” onde terá o valor lógico **V** somente se ambas as proposições forem verdadeiras, caso contrário será obtido o valor **F**. Normalmente é representada pelo conectivo “e” e pela notação “ $s \wedge t$ ”, por exemplo:

- $s \wedge t$  : O Sol é amarelo e o mar é salgado (**V**)
- $r \wedge p$  : A Lua é um satélite e Vênus é uma estrela (**F**).

Na operação de disjunção - inclusiva - entre duas proposições “s” e “t”, será obtido o valor lógico (V) caso pelo menos uma das proposições seja verdade, e obterá o valor (F) caso ambas as proposições sejam falsas. Ela pode ser representada pelo conectivo “ou” e pela notação “ $s \vee t$ ”, por exemplo:

- $s \vee t$  :  $\pi > 2$  ou Londres é capital do Japão (**V**)
- $r \vee p$  : Abraham Lincoln nasceu no Brasil ou  $\sqrt{-1} = 1$  (**F**)

Em uma operação condicional “se s então t”, o valor lógico é falso caso “s” seja (V) e “t” seja (F), nas demais situações ela será verdadeira. A condicional é representada pelo conectivo “se... então...” e também pela notação “ $s \rightarrow t$ ”, por exemplo:

- $s \rightarrow t$  : Se abril tem 30 dias então a Terra é plana (**F**)
- $r \rightarrow p$  : Se Santos Dumont é paraguaio então o ano tem 9 meses (**V**)

No tipo bicondicional temos uma operação de duas proposições “s” e “t”, onde será obtido o valor (V) caso ambos os valores sejam verdadeiros, ou ambos sejam falsos, caso contrário será obtido (F). É representada pelo conectivo “se e somente se” e pela notação “ $s \leftrightarrow t$ ”, por exemplo:

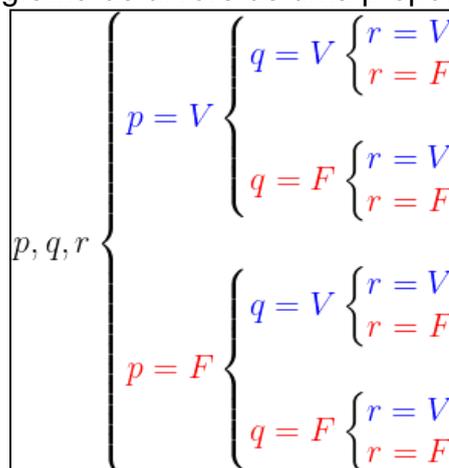
- $s \leftrightarrow t$  : Natal é a capital do RN se e somente se a neve for branca (V) 12
- $r \leftrightarrow p$  : Os pássaros não voam se e somente se os peixes não nadam (V)
- $q \leftrightarrow f$  : 3 é par se e somente se 5 for ímpar (F)

### 3.1.3. Tabelas-Verdade

A tabelas-verdade é uma ferramenta utilizada para estudar a lógica matemática. Através dela, é possível determinar qual será o valor lógico de uma proposição composta, ou seja, saber se a fórmula é verdadeira ou falsa. “O valor lógico de qualquer proposição composta depende unicamente dos valores lógicos das proposições simples componentes, ficando por eles univocamente determinado” (FILHO, 1975, p. 14).

Para calcular o número de linhas de uma tabela-verdade é necessário saber a quantidade de proposições simples que compõem a fórmula. Retomando o princípio do terceiro excluído, cada proposição simples pode assumir dois valores distintos, verdadeiro ou falso, e quanto maior o número de proposições simples maior é o número de combinações possíveis entre seus resultados, que aumentam de forma exponencial como na Figura 4.

Figura 4 - Diagrama de árvore de uma proposição composta



Fonte: Método Papio: O sábio da matemática (2020).

A fórmula usada para calcular o número de linha da tabela, baseando se no estudo da Análise Combinatória, é  $2^n$ , sendo “n” o número de proposições simples  $n$  que fazem parte da proposição composta contemplada pela tabela-verdade.

Tendo em mãos os números de linhas da tabela, e sabendo que a quantidade 13 de colunas é igual ao número de proposições simples mais a coluna do resultado, podemos montar uma tabela como a Tabela 2.

Tabela 2 – 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 (2021).

### 3.1.4. Tautologia, Contradição e Contingência

Ainda conforme (FILHO, 1975) em sua obra, existem alguns tipos especiais de proposições que precisam ser destacadas. Estas são as tautologias, as contradições e as contingências.

Tautologia é uma proposição composta cujo valor lógico sempre será verdadeiro, ou seja, o valor lógico da proposição “ $Q(g, a, p, \dots)$ ”, mostrado na última coluna da tabelaverdade, sempre será verdadeira independentemente dos valores de “ $g$ ”, “ $a$ ”, “ $p$ ”, “etc”, assim como o exemplo da Tabela 3.

Tabela 3 - Tabela-verdade de uma tautologia

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

Fonte: FILHO, 1975.

Já a contradição, ao contrário da tautologia, é uma fórmula proposicional a qual o valor lógico encontrado sempre será falso. Em outras palavras, uma proposição composta  $J(l, m, n, \dots)$  que tenha “Falso” como valor lógico, independentemente dos valores de “l”, “m”, “n” e “etc”, é uma contradição. Desse modo, pode-se inferir que uma contradição é uma **negação** uma tautologia.

Tabela 4 - Tabela-verdade de uma contradição

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

Fonte: FILHO, 1975.

Por fim, uma contingência é uma proposição composta cuja a última coluna da tabela-verdade mostra os valores V (verdadeiro) e F (falso) pelo menos uma vez, como na Tabela 5. Desse modo, temos que uma contingência é uma proposição que não é nem contradição nem tautologia.

Tabela 5 - Tabela-verdade de uma contingência (Negação)

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

Fonte: autoria própria (2023).

### 3.1.5. Implicação Lógica

Uma proposição composta “ $P(p, q, r, \dots)$ ” implica logicamente uma proposição “ $Q(p, q, r, \dots)$ ” ( $P \Rightarrow Q$ ) sempre quando “ $P \rightarrow Q$ ” for uma tautologia. Em outros termos, teremos uma implicação se em todos os casos em que “ $P$ ” for verdadeiro “ $Q$ ” também for verdadeiro (FILHO, 1975, p. 49).

Existem algumas propriedades que a implicação segue. A primeira delas é a propriedade reflexiva (R), que determina que toda proposição  $P(p, q, r, \dots)$  implica logicamente nela mesma ( $P \Rightarrow P$ ), isso é evidente já que como as fórmulas são idênticas os valores “V” vão sempre incidir. A segunda propriedade é a transitiva (T), que diz que “se  $P(p, q, r, \dots) \Rightarrow Q(p, q, r, \dots)$  e  $Q(p, q, r, \dots) \Rightarrow R(p, q, r, \dots)$ , então  $P(p, q, r, \dots) \Rightarrow R(p, q, r, \dots)$ ”, esse é um princípio da lógica matemática clássica e pode ser comparado com “ $x = y$  e  $y = z$  então  $x = z$ ”.

Além das propriedades, a implicação lógica também possui regras de inferência, que delimitam as operações usadas para encontrar preposições 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, ou  $P(p, q, r, \dots) \Leftrightarrow Q(p, q, r, \dots)$  quando a última linha, ou resultados, das tabelas de “P” e “Q” forem iguais. Em outras palavras, a proposição “P” é equivalente a “Q” se  $P \leftrightarrow Q$  é uma tautologia (FILHO, 1975, p. 55).

Assim como a Implicação, a Equivalência lógica possui algumas propriedades. Ela possui a propriedade reflexiva (R) e a transitiva (T) ( $P \Leftrightarrow P$  / Se  $P \Leftrightarrow Q$  e  $Q \Leftrightarrow R$ , então  $P \Leftrightarrow R$ ), mas, além dessas duas, também existe a simetria (S), que fala que “se  $P \Leftrightarrow Q$  então  $Q \Leftrightarrow P$ ”, pois, diferentemente da implicação, os “V” de “P” precisam coincidir com os de “Q” e viceversa.

Além de propriedades, existem algumas regras para a equivalência lógica.

Entre elas podemos citar 3:

- 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

Os diferentes tipos de proposições possuem diferentes propriedades matemáticas incorporadas à chamada Álgebra das Proposições. (FILHO, 1975) - no livro Iniciação à Lógica Matemática - lista as propriedades da conjunção, as propriedades da disjunção, as propriedades da conjunção e da disjunção, a negação da condicional e a negação 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 ao mesmo tempo verdadeiras equivale a afirmar que uma pelo menos é falsa.
- 2) Negar que uma pelo menos de duas proposições é verdadeira equivale a afirmar que ambas são falsas. (FILHO, 17 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

Apesar do “Método das tabelas-verdade” ser a forma mais comum de representar uma implicação ou equivalência lógica, ele não é o único. O “Método dedutivo” é uma outra maneira de demonstrar, de forma eficiente, as equivalências da “Álgebra das Proposições”.

Como exemplo, (FILHO, 1975) considera uma proposição simples “p”, uma proposição com valor falso “c” e uma proposição com valor verdadeiro “t”, onde “c ⇒ p” ou “p ⇒ t”. Tendo então, respectivamente:

- $c \rightarrow p \Leftrightarrow \sim c \vee p \Leftrightarrow t \vee p \Leftrightarrow t$
- $p \rightarrow t \Leftrightarrow \sim p \vee t \Leftrightarrow t$

Ao observar as tabelas-verdade “c → p” e “p → t”, na Tabela 6 é mostrado que tais condicionais são tautológicas.

Tabela 6 - Tabelas-verdade de “c → p” e “p → t”

<i>p</i>	<i>c</i>	<i>t</i>	<i>c</i> → <i>p</i>	<i>p</i> → <i>t</i>
V	F	V	V	V
F	F	V	V	V

Fonte: FILHO, 1975.

No Método dedutivo é possível reduzir a quantidade conectivos usados em uma expressão. Dos cinco conectivos principais (“~”, “∧”, “∨”, “→” e “↔”) três podem se apresentar com os termos de dois dos pares a seguir:

- 1) “~” e “∨”
- 2) “~” e “∧”
- 3) “~” e “→”

Os efeitos dessas reduções são apresentados na Figura 5. Ainda vale ressaltar que os conectivos “ $\wedge$ ”, “ $\vee$ ”, “ $\rightarrow$ ” não podem ser expressos por meio dos termos “ $\sim$ ” e “ $\leftrightarrow$ ”.

Figura 5 - Exemplos de redução dos números de conectivos

(1)  $\wedge, \vee, \leftrightarrow$  exprimem-se em termos de  $\sim$  e  $\vee$  :

$$p \wedge q \leftrightarrow \sim \sim p \wedge \sim \sim q \leftrightarrow \sim (\sim p \vee \sim q)$$

$$p \vee q \leftrightarrow \sim \sim p \vee \sim \sim q \leftrightarrow \sim (\sim p \wedge \sim q)$$

$$p \leftrightarrow q \leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p) \leftrightarrow \sim (\sim (\sim p \vee q) \vee \sim (\sim q \vee p))$$

(2)  $\vee, \rightarrow, \leftrightarrow$  exprimem-se em termos de  $\sim$  e  $\wedge$  :

$$p \vee q \leftrightarrow \sim \sim p \vee \sim \sim q \leftrightarrow \sim (\sim p \wedge \sim q)$$

$$p \rightarrow q \leftrightarrow \sim p \vee q \leftrightarrow \sim (p \wedge \sim q)$$

$$p \leftrightarrow q \leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p) \leftrightarrow \sim (p \wedge \sim q) \wedge \sim (\sim p \wedge q)$$

(3)  $\wedge, \vee$  e  $\leftrightarrow$  exprimem-se em termos de  $\sim$  e  $\rightarrow$  :

$$p \wedge q \leftrightarrow \sim (\sim p \vee \sim q) \leftrightarrow \sim (p \rightarrow \sim q)$$

$$p \vee q \leftrightarrow \sim \sim p \vee q \leftrightarrow \sim p \rightarrow q$$

$$p \leftrightarrow q \leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p) \leftrightarrow \sim ((p \rightarrow q) \rightarrow \sim (q \rightarrow p))$$

Fonte: FILHO, 1975.

Com isso, ainda consoante a (FILHO, 1975), é dito que uma proposição se encontra em sua Forma Normal (FN) quando contém apenas os conectivos “ $\sim$ ”, “ $\wedge$ ”, e “ $\vee$ ” em sua estrutura. Dessa forma, ainda é possível dividir a FN de uma proposição em duas subcategorias: a Forma Normal Conjuntiva (FNC) e a Forma Normal Disjuntiva (FND).

Em uma proposição na Forma Normal Conjuntiva o “ $\sim$ ” só pode incidir em letras proposicionais, ou seja, não pode aparecer de forma repetida nem sobre outros conectivos. Além disso, o conectivo “ $\vee$ ” não pode ter alcance ao “ $\wedge$ ”, logo não existem arranjos como “ $p \vee (q \wedge r)$ ”.

Para montar uma FNC equivalente a uma proposição remova desta os conectivos “ $\rightarrow$ ” e “ $\leftrightarrow$ ”, por meio da substituição de “ $p \rightarrow q$ ” por “ $\sim p \vee q$ ” e “ $p \leftrightarrow q$ ” por “ $(\sim p \vee q) \wedge (p \vee \sim q)$ ”, retirando as negações que são duplas ou prescindindo parênteses por meio das regras da “Dupla Negação” e “de Morgan” e por fim substituir “ $p \vee (q \wedge r)$ ” e “ $(p \wedge q) \vee r$ ” por suas expressões equivalentes: “ $(p \vee q) \wedge (p \vee r)$ ” e “ $(p \vee r) \wedge (q \vee r)$ ” respectivamente.

Tal como a FNC, uma FND não apresenta os conectivos “ $\rightarrow$ ” e “ $\leftrightarrow$ ” e o “ $\sim$ ” só incide em letras proposicionais. Entretanto, diferentemente da FN anterior, na FND será o conectivo “ $\wedge$ ” que não terá alcance ao “ $\vee$ ”, logo não existem arranjos como “ $p \wedge (q \vee r)$ ” nessa forma.

Sempre é possível formar uma FND equivalente a uma proposição removendo os conectivos “ $\rightarrow$ ” e “ $\leftrightarrow$ ” mediante substituição, removendo as negações que são duplas ou prescindindo parênteses por meio das regras da “Dupla Negação” e “de Morgan” e por fim substituir “ $p \wedge (q \vee r)$ ” e “ $(p \vee q) \wedge r$ ” por suas expressões equivalentes: “ $(p \wedge q) \vee (p \wedge r)$ ” e “ $(p \wedge r) \vee (q \wedge r)$ ” respectivamente.

## 3.2. CONCEITOS SOBRE ALGORITMOS

### 3.2.1. Conceitos Fundamentais

Múltiplos autores apresentam diferentes definições para o que é um algoritmo, contudo, de forma geral, um algoritmo pode ser descrito como uma sequência finita de instruções lógicas que devem ser seguidas para cumprir um objetivo predeterminado.

(ASCENCIO & CAMPOS) - apresenta alguns exemplos simples de algoritmos, como o mostrado na Figura 6.

Figura 6 - Algoritmo para somar três números (Descrição Narrativa)

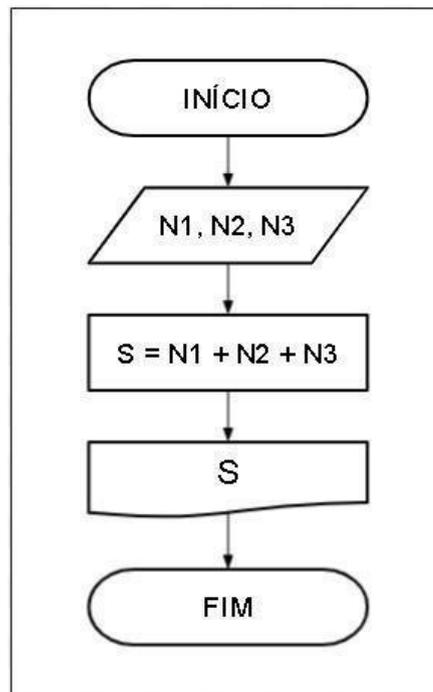
<b>SOMAR TRÊS NÚMEROS</b>	
<b>PASSO 1</b>	– Receber os três números.
<b>PASSO 2</b>	– Somar os três números.
<b>PASSO 3</b>	– Mostrar o resultado obtido.

Fonte: FPC (2016)

(ASCENCIO & CAMPOS) ainda descrevem os diferentes tipos de algoritmos. O primeiro deles é a descrição narrativa, na qual os algoritmos são escritos em uma linguagem natural, como na figura 6.

O segundo tipo de representação de algoritmo é o fluxograma, que consiste em fazer uso de símbolos gráficos predefinidos para escrever o passo a passo para a solução do problema trabalhado, tal como mostrado na Figura 7.

Figura 7 - Algoritmo para somar três números (Fluxograma)



Fonte: autoria própria (2021)

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, tal como exemplificado na Figura 8, em que é usado o pseudocódigo do Português.

Figura 8 - Algoritmo para somar três números (Pseudocódigo no Português)

```
programa
{
  Funcao inicio()
  {
    inteiro n1, n2, n3 , soma

    escreva("Digite o primeiro número: ")
    leia(n1)

    escreva("Digite o segundo número: ")
    leia(n2)

    escreva("Digite o terceiro número: ")
    leia(n3)

    soma = n1 + n2 +n3 // Soma os três valores

    escreva("\nA soma dos números é igual a: ", soma) // Exibe o resultado da soma
  }
}
```

Fonte: autoria própria (2023)

### 3.2.2. Tipos Primitivos de Dados

Dados são, de forma geral, o registro de uma informação específica. Na programação os dados possuem classificações variadas, contudo, existem alguns tipos primitivos presentes na maioria dos algoritmos. Estes são os numéricos, os lógicos e os literais.

Os dados numéricos são divididos entre os inteiros e reais. Os inteiros podem ser compreendidos como variáveis que contêm quaisquer números que pertençam ao grupo dos inteiros, ou seja, podem ser valores positivos, negativos ou nulo que não possuem casas decimais. Paralelamente a isso, os reais armazenam números reais na forma de frações decimais, podem ser positivos ou negativos, são separados por pontos e diferente dos inteiros estes possuem casa decimal. (CITAÇÃO)

Os dados do tipo lógico são os mais simples dos tipos primitivos. Também chamadas de booleanas, as variáveis lógicas podem assumir valores de verdadeiro ou falso, assim como as proposições lógicas.

Os dados do tipo literal, tal como o numérico, podem ser subclassificados com o tipo caractere, que armazena um único caractere, ou como do tipo cadeia, que armazena um conjunto de caracteres que podem formar palavras ou frases. Ambos podem armazenar letras maiúsculas, letras minúsculas, números (não podem ser utilizados em cálculos) e caracteres especiais.

### **3.2.3. Operadores: Aritméticos, Lógicos e Relacionais**

Os operadores são elementos fundamentais, componentes das expressões, que são muito usadas para a construção de algoritmos. Os operadores se dividem em vários tipos, entre eles é válido destacar os Aritméticos, Lógicos e Relacionais.

Os operadores aritméticos são usados em expressões matemáticas, ou seja, são usados para realizar cálculos. Os operadores básicos, disponíveis na maioria das linguagens de programação, estão sendo exibidos na Tabela 7.

Tabela 7 – Operadores aritméticos básicos.

<b>Operação</b>	<b>Símbolo</b>	<b>Prioridade</b>
Adição	+	1
Subtração	-	1
Multiplicação	*	2
Divisão	/	2
Resto da divisão inteira	%	2

Fonte: Linguagem Portugol (2013).

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. Os operadores básicos são o da conjunção, da disjunção e da negação, mostrados na Tabela 8.

Tabela 8 – Operadores lógicos básicos.

<b>Operador</b>	<b>Prioridade</b>
ou	1
e	2
nao	3

Fonte: Linguagem Portugol (2013).

Os operadores relacionais são utilizados quando se quer verificar ou quando se quer comparar os valores de variáveis e de outras expressões. Eles emitem como resultado valores lógicos, verdadeiro ou falso, e a maioria é usada com valores numéricos. Os principais operadores relacionais estão listados na Tabela 9.

Tabela 9 – Operadores relacionais básicos.

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

Fonte: Linguagem Portugol (2013).

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 7), 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

Quando se está trabalhando com algoritmos é comum, e constante, a criação de variáveis para armazenar conteúdo. Para determinar ou alterar o conteúdo salvo dentro de uma variável são usados os comandos de atribuição, entre eles o mais importante é o "=", já que a maioria dos outros comandos de atribuição são derivados do comando supracitado.

O comando de entrada é utilizado para solicitar dados ao usuário, dados estes que, depois de recebidos, são guardados em variáveis. Esse comando é representado no portugol como "leia".

Já o comando de saída é utilizado para mostrar informações ao usuário, que podem ser mensagens ou dados produzidos pelo algoritmo. Esse comando é representado no portugol como "escreva".

#### 3.2.5. Estruturas Condicionais

As estruturas condicionais são usadas quando o programador deseja estabelecer condições para a execução de uma determinada sequência de códigos. Se a condição for

verdadeira, um bloco de comandos é executado, caso seja falsa os comandos condicionados são pulados.

As condições usadas nessas estruturas assumem a forma de expressões, que podem utilizar, além de números e variáveis, os operadores aritméticos, lógicos e, principalmente, os relacionais em sua formação.

Figura 9 - Algoritmo verificador de nota no Portugol (pseudocódigo)

```
programa
{
    funcao inicio ()
    {
        real nota

        escreva("Informe sua nota: ")
        leia(nota)

        se (nota >= 6)
        {
            escreva("Você foi aprovado")
        }
        senao
        {
            escreva("Você foi reprovado")
        }
    }
},
```

Fonte: autoria própria (2023).

### 3.2.6. Estruturas de Repetição

Existem situações em que é necessário repetir várias vezes uma mesma sequência de códigos. Quando isso ocorre um número muito grande de vezes ou quando a quantidade de repetições pode variar dependendo do usuário é feito o uso de uma estrutura de repetição.

De forma similar às estruturas condicionais, uma estrutura de repetição exige uma condição para que a execução da sequência de códigos continue se repetindo. Essa condição também será uma expressão, e enquanto o valor lógico dessa expressão for verdadeiro a repetição continuará.

Figura 10 - Algoritmo para montar tabuada no Portugol (Pseudocódigo)

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

Fonte: Biblioteca Portugal, 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. Google Classroom

Foi criada uma sala de aula no Google Classroom para a execução da monitoria. Sendo ela destinada para a comunicação entre os monitores e o orientador da disciplina de FLA, permitindo o envio de mensagens, agendamento de reuniões ao longo da monitoria e também a publicação de listas de exercícios durante o primeiro de aula do ano de 2022.

#### 4.1.2. Quadro Branco

O quadro branco pode ser caracterizado como uma ferramenta para a ajuda na explicação do conteúdo, tendo em vista que ele é: uma superfície lisa de cor branca, geralmente feita de laminado melamínico.

#### 4.1.3. Marcador (piloto)

O marcador é um tipo especial de caneta hidrográfica, desenvolvido principalmente para o uso em quadro branco, sua tinta é facilmente removida da superfície do quadro por um apagador de lã e pode ser encontrado de várias cores no mercado.

#### **4.1.4. PORTUGOL STUDIO**

O aplicativo do Portugol *Studio* foi utilizado para criar e executar alguns códigos de exemplo aos alunos, além de auxiliar nos momentos de tiragem de dúvidas quanto às atividades passadas pelo professor.

### **4.2. REUNIÕES DA MONITORIA**

#### **4.2.1. REUNIÕES DE ORIENTAÇÃO**

Foram realizadas reuniões entre os monitores e o orientador e professor da disciplina de FLA. e nesses encontros eram informadas, o que era para ser feito durante toda a monitoria, passados os conteúdos que seriam aplicados durante a aula do professor.

#### **4.2.2. ENCONTRO COM ALUNATO**

Os encontros eram realizados toda semana, de segunda-feira a sexta-feira, no seguinte horário: de 8:30 às 11:30. Na localidade da biblioteca do campus, que fica localizada no primeiro piso do prédio principal do nosso campus.

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

### **5. CARACTERIZAÇÃO DAS ATIVIDADES DA MONITORIA**

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

#### **5.1. DISCIPLINA-OBJETO**

A disciplina objeto desta monitoria é a disciplina de Fundamentos de Lógica e Algoritmos, um dos componentes curriculares para se tornar técnico em informática. Ela possui uma carga horária de 75 horas-relógio, o que equivale a 100 horas-aula, e tem a função de apresentar os conceitos, métodos e técnicas que guiam a construção de algoritmos.

Ela tem como objetivos permitir que os discentes sejam capazes de compreender a lógica da programação, elaborar algoritmos, conhecer as estruturas básicas de dados,

compreender e saber utilizar os operadores lógicos da álgebra booleana e interpretar problemas de lógica proposicional.

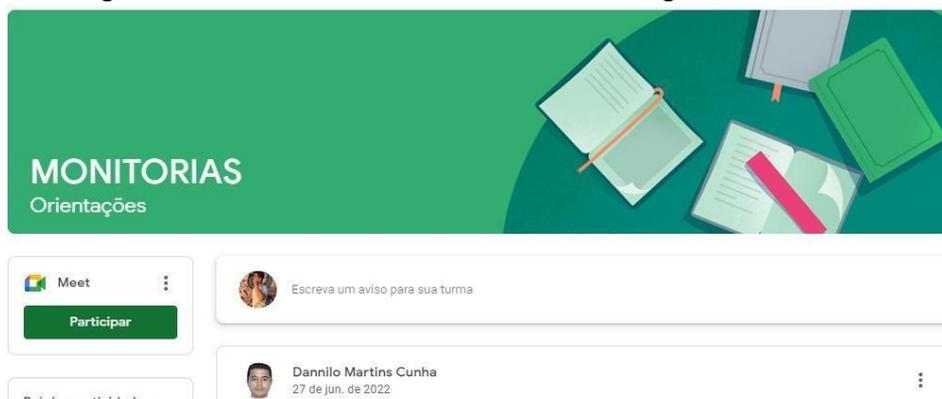
As bases científico-tecnológicas, também conhecidas como conteúdos, são:

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

## 5.2. DESCRIÇÃO DAS FERRAMENTAS UTILIZADAS

O Google Classroom, ou Google Sala de Aula, é uma plataforma de ensino/aprendizagem online que permite o gerenciamento de uma ou mais turmas de alunos por um ou mais professores. Ela está integrada a um conjunto de ferramentas digitais que permite a criação de reuniões, publicação de materiais, organização de atividades, etc.

Figura 11 - Interface de uma sala do Google Classroom



Fonte: autoria própria (2021).

O Portugal *Studio* é uma plataforma de pseudocódigo completamente em português, criada para pessoas iniciantes em programação interessadas em aprender os conceitos de algoritmos.



Fonte: autoria própria (2021).

### 5.3. RESULTADOS OBTIDOS

A monitoria deu uma oportunidade de pôr em prática e exercitar os conhecimentos adquiridos ao longo do curso de informática. Além disso, ela permitiu a aquisição de novos conhecimentos e experiências quanto à área de docência e gestão organizacional.

Contudo, apesar dos esforços dos monitores e do professor, houve um resultado negativo observado ao longo da prática: a ausência dos alunos ao longo da monitoria.

## 6. ANÁLISE E DISCUSSÃO DOS RESULTADOS

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

### 6.1. AQUISIÇÃO DE CONHECIMENTOS E EXPERIÊNCIAS

Com o decorrer da monitoria, foi possível aprender mais e adquirir Experiências sobre as áreas acadêmicas da docência e da gestão organizacional. Pesquisar e ler materiais para se aprofundar nos conteúdos da disciplina e poder auxiliar os alunos, organizar horários para estar disponível para as dúvidas dos discentes e se comprometer com eles foram alguns dos costumes necessários, obtidos ao longo desta prática por parte dos monitores.

Ademais, trocar conhecimentos e experiências com o professor 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.

### 6.2. 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. Apesar da comunicação via *WhatsApp* por parte do professor, da presença dos alunos no ambiente virtual criado no *Google Classroom*, como mostrado na Figura 15, e de algumas ocasionais mensagens para os monitores, não ocorreu uma participação efetiva do alunado nas reuniões organizadas na sede do nosso campus.

Figura 15 - Página de participantes da turma do *Classroom*.

## Professores

---



Dannilo Martins Cunha



## Colegas de turma

---

3 alunos



Luiz Felipe das chagas Bezerra



Ana Karollini de Melo Silva



Fonte: autoria própria (2023).

A seguir, tem-se alguns dos fatores que podem ter contribuído com tal ausência:

- O trancamento de matrícula;
- Cancelamento de matrícula;

## **7. CONSIDERAÇÕES FINAIS**

No presente documento, foi apresentado o relatório do desempenho da monitoria de Fundamentos de Lógica e Algoritmo no IFRN Campus Avançado Lajes, que teve como objetivo auxiliar o professor, sanando as dúvidas dos discentes e realizar a atividade da prática profissional obrigatória ao final do curso técnico em informática.

Tal prática profissional foi muito importante, pois, além de ter como objetivo auxiliar os iniciandos da área de informática, ela permitiu aos monitores voluntários executar atividades relacionadas com o ambiente acadêmico, proporcionou o enriquecimento da formação e do currículo profissional dos alunos, estimulou a cooperação entre estudante e professor e promoveu a troca de conhecimento.

Ademais, as atividades desempenhadas permitiram que o monitor aprimorasse e adquirisse 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, houve um resultado negativo observado ao longo da prática: a ausência dos alunos no decorrer de tal prática.

Em suma, percebe-se que os objetivos da monitoria foram parcialmente atingidos. As atividades permitiram o desenvolvimento do monitor como profissional, tendo em vista o comprometimento e dedicação necessários ao longo da laboração. Todavia, graças a ausência observada por parte do alunado, pode-se considerar que houve uma falha no desenvolvimento da prática. Mas, é válido ressaltar que, como nem docentes e nem monitores têm controle sobre o fluxo de requisição dos alunos, esse resultado já podia ser esperado e também agrega a formação do profissional.

## REFERÊNCIAS

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

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

MOTA, Cláudia. Aprenda a determinar o número de linhas de uma tabela verdade. Método Papio, 14 de julho, 2020. Disponível: . Acesso em: 27 set. 2021.

<b>Dados do Relatório Científico</b>	
Título e subtítulo: Relatório da Prática Profissional: Monitoria de Fundamentos de Lógica e Algoritmos	
Tipo de relatório: Monitoria	Data: 16/02/2023
Título do projeto/ programa/ plano: Monitoria de Fundamentos de Lógica e Algoritmos	
Autor: Natanael Asafe Jales da Costa	
Instituição e endereço completo: Instituto Federal do RN Campus Lajes Endereço: BR-304, Km 120, s/n - Centro, Lajes - RN, 59535-000	
Resumo: <p>Esta prática profissional em questão foi caracterizada na forma de uma monitoria, que teve como objetivo auxiliar os novos alunos da área de informática e permitiu aos monitores voluntários executar atividades relacionadas com o ambiente acadêmico, enriquecendo a formação do currículo profissional dos mesmos. Tais objetivos foram alcançados através de reuniões regulares com o auxílio de ferramentas digitais e encontros físicos. Percebeu-se, então, que os objetivos foram em parte atingidos, pois permitiram o desenvolvimento do monitor como profissional, apesar da falha no desenvolvimento da prática causada pela ausência dos alunos do curso de informática. Contudo, é válido lembrar que, como não se têm controle sobre o fluxo e da procura dos alunos, esse tal resultado poderia ser esperado, com tudo isso ainda agregou à formação do monitor. Por fim, os resultados obtidos com a prática foram a aquisição e aprimoramento dos conhecimentos acadêmicos, ganho de experiências em organização, disciplina e comprometimento.</p>	
Palavras-chave: Monitoria. Lógica. Algoritmos. Prática Profissional.	

Nº de páginas: 36 páginas	
Jornada de trabalho: 3 Horas Diárias	Horas semanais: 15 Horas
Total de horas: 360 Horas	

Observações/notas
-------------------