

**Relatório de Atividade Prática**

**LABORATÓRIO #3**

**Integrantes do Grupo**

Nome: Anderson Correa Nicodemo	TIA: 32285671
Nome: Felipe do Nascimento Fonseca	TIA: 42215366
Nome: Giovani Alves Lavia	TIA: 42218365
Nome: Gustavo Garabetti Munhoz	TIA: 42211956

**EXERCÍCIO 1**

**CÓDIGO FONTE**

```
1 .data
2     vetor: .word 2, -5, 12, 7, -3, 99, 8, 54, 21, -45, 67, 61, 55
3     tamanho: .word 13
4     colchete: .asciiz "]"
5     virgula: .asciiz ", "
6     saidaA: .asciiz "O vetor original é: ["
7     saidaB: .asciiz "O maior elemento do vetor é: "
8     saidaC: .asciiz "\nO menor elemento do vetor é: "
9     saidaD: .asciiz "\nA média dos elementos do vetor é: "
10    saidaE: .asciiz "\nDigite um número: "
11    encontrado: .asciiz "Número encontrado no vetor"
12    naoEncontrado: .asciiz "Número não encontrado no vetor"
13
14 .text
15     j main
16
17     imprime_original: # Imprimir o vetor original
18
19     move $t9, $ra
20     jal carrega # carrega o vetor ($s0) e tamanho ($t0)
21
22     li $v0, 4
23     la $a0, saidaA
24     syscall
25
26     loop:
27
28     lw $s1, 0($s0)
29
30     li $v0, 1
31     move $a0, $s1
32     syscall
33
34     add $s0, $s0, 4
35     add $t1, $t1, 1
36
37     beq $t1, $t0, fim_loop
38
39     li $v0, 4
40     la $a0, virgula
41     syscall
42
43     j loop
44
45     fim_loop:
46
47     li $v0, 4
48     la $a0, colchete
49     syscall
50
51     jr $t9
```

```

53     imprime_maior: # Encontrar e imprimir o maior elemento dentro do vetor
54
55     move $t9, $ra
56     jal carrega # carrega o vetor ($s0) e tamanho ($t0)
57
58     lw $t3, 0($s0)
59
60     # while( $t1 < tamanho )
61     while_01:
62     addi $s0, $s0, 4
63     lw $t2, 0($s0)
64     bge $t3, $t2, final_while01
65
66     move $t3, $t2
67
68     final_while01:
69     addi $t1, $t1, 1
70     blt $t1, $t0, while_01
71
72     li $v0, 4
73     la $a0, saidaB
74     syscall
75
76     li $v0, 1
77     move $a0, $t3
78     syscall
79
80     jr $t9
81
82
83     imprime_menor: # Encontrar e imprimir o menor elemento dentro do vetor
84
85     move $t9, $ra
86     jal carrega # carrega o vetor ($s0) e tamanho ($t0)
87
88     lw $t3, 0($s0)
89
90     # while( $t1 < tamanho )
91     while_02:
92     addi $s0, $s0, 4
93     lw $t2, 0($s0)
94     ble $t3, $t2, final_while02
95
96     move $t3, $t2
97
98     final_while02:
99     addi $t1, $t1, 1
100    blt $t1, $t0, while_02
101
102    li $v0, 4
103    la $a0, saidaC
104    syscall
105
106    li $v0, 1
107    move $a0, $t3
108    syscall
109
110    jr $t9
111
112    imprime_media: # Calcular e imprimir a média dos elementos do vetor
113
114    move $t9, $ra
115    jal carrega # carrega o vetor ($s0) e tamanho ($t0)
116
117    lw $t3, 0($s0)
118
119    # while( $t1 < tamanho )
120    while_03:
121    addi $s0, $s0, 4
122    lw $t2, 0($s0)
123    add $t3, $t3, $t2
124
125    #final do while
126    addi $t1, $t1, 1
127    blt $t1, $t0, while_03
128
129    li $v0, 4
130    la $a0, saidaD
131    syscall

```

```

133     li $v0, 1
134     div $a0, $t3, $t0
135     syscall
136
137     jr $t9
138
139     verifica_elemento: # Verificar se um elemento está presente no vetor
140
141     move $t9, $ra
142     jal carrega # carrega o vetor ($s0) e tamanho ($t0)
143     addi $t0, $t0, 1
144
145     lw $t3, 0($s0)
146
147     li $v0, 4
148     la $a0, saidaE
149     syscall
150
151     li $v0, 5
152     syscall
153
154     # while( $t1 < tamanho )
155     while_04:
156     beq $v0, $t3, Achou
157     addi $s0, $s0, 4
158     lw $t3, 0($s0)
159
160     #final do while
161     addi $t1, $t1, 1
162     blt $t1, $t0, while_04
163
164     li $v0, 4
165     la $a0, naoEncontrado
166     syscall
167
168     jr $t9
169
170     Achou:
171
172     li $v0, 4
173     la $a0, encontrado
174     syscall
175
176     jr $t9
177
178
179     carrega: # carrega as variaveis para evitar que estejam manipuladas ao começo de cada função
180
181     la $s0, vetor
182     lw $t0, tamanho
183     li $t1, 0
184
185     jr $ra
186
187     main:
188
189     jal imprime_original
190     jal imprime_maior
191     jal imprime_menor
192     jal imprime_media
193     jal verifica_elemento
194
195     li $v0, 10
196     syscall
197
198

```

## TESTES DE FUNCIONALIDADE

```
O vetor original é: [2, -5, 12, 7, -3, 99, 8, 54, 21, -45, 67, 61, 55]
O maior elemento do vetor é: 99
O menor elemento do vetor é: -45
A média dos elementos do vetor é: 25
Digite um número: 99
Número encontrado no vetor
-- program is finished running --

O vetor original é: [2, -5, 12, 7, -3, 99, 8, 54, 21, -45, 67, 61, 55]
O maior elemento do vetor é: 99
O menor elemento do vetor é: -45
A média dos elementos do vetor é: 25
Digite um número: 2
Número encontrado no vetor
-- program is finished running --

O vetor original é: [2, -5, 12, 7, -3, 99, 8, 54, 21, -45, 67, 61, 55]
O maior elemento do vetor é: 99
O menor elemento do vetor é: -45
A média dos elementos do vetor é: 25
Digite um número: 55
Número encontrado no vetor
-- program is finished running --

O vetor original é: [2, -5, 12, 7, -3, 99, 8, 54, 21, -45, 67, 61, 55]
O maior elemento do vetor é: 99
O menor elemento do vetor é: -45
A média dos elementos do vetor é: 25
Digite um número: 68
Número não encontrado no vetor
-- program is finished running --

O vetor original é: [2, -5, 12, 7, -3, 99, 8, 54, 21, -45, 67, 61, 55]
O maior elemento do vetor é: 99
O menor elemento do vetor é: -45
A média dos elementos do vetor é: 25
Digite um número: 13
Número não encontrado no vetor
-- program is finished running --
```

## COMENTÁRIOS GERAIS

Nesse exercício eu precisei percorrer a array diversas vezes, e para isso decidi deixar na primeira função um loop para que ficasse mais evidente o que está acontecendo no programa, nas outras funções apliquei versões de loop semelhantes.

É possível ver que dentro de todas as funções eu chamei outra função, a “Carrega”, foi a primeira vez que implementei uma função dentro de outra, e notei que não era possível mais usar o valor do registrador \$ra, pois era sobrescrito pelo endereço da nova função. Para contornar isso todas às vezes antes de chamar a carrega eu passava o valor de \$ra para \$t9, dessa forma o jr chamava o valor dentro de \$t9 para voltar a main.

A formatação foi uma das partes mais fáceis, apenas chamando as syscalls alternadamente para formar o resultado.

Para encontrar um número no vetor, eu o percorri e caso encontrasse antes do loop acabar, ele pulava para um trecho do código que imprime a mensagem de encontrado, caso não encontrasse, saíria do loop e já iria de encontro a um trecho de código que imprime a mensagem de não encontrado.

## EXERCÍCIO 2

### CÓDIGO FONTE

```
1 .data
2 Notas_Alunos: .space 40 #Memoria alocada para notas (vetor de 10 elementos x 4 bytes de cada nota)
3 Contador_Alunos: .word 0 #Contador de alunos para mensagem
4 Contador_Aprovados: .word 0 #Contador de alunos aprovados
5 Contador_Reprovados: .word 0 #Contador de alunos reprovados
6 Contador_Zeros: .word 0 #Contador de alunos zerados
7 #Mensagens de sistema:
8 Titulo: .asciiz "Sistema de Notas:\n"
9 Msg_Solicitar1: .asciiz "Digite a nota do "
10 Msg_Solicitar2: .asciiz "" Aluno: "
11 Msg_Notas: .asciiz "\nNotas dos alunos: "
12 Msg_Aprovados: .asciiz "\nA quantidade de alunos aprovados: "
13 Msg_Reprovados: .asciiz "\nA quantidade de alunos reprovados: "
14 Msg_Zerados: .asciiz "\nA quantidade de alunos com nota igual a zero: "
15 Simbolo_Chave1: .asciiz "Vetor de notas: ["
16 Simbolo_Chave2: .asciiz "]"
17 Virgula: .asciiz ", "
18 Msg_Erro: .asciiz "Nota invalida, digite uma nota maior que zero e menor que dez: "
19
20 .text
21 Main:
22 #Inicializa o programa
23 li $t0, 0 #Registrador para guardar o indice vetor de notas que inicia em 0
24 li $t1, 10 #Quantidade de alunos
25 la $t2, Notas_Alunos #Endereço do vetor de notas
26 li $t3, 6 #Variavel de quem foi aprovado (maior que 6) para comparação
27
28 li $v0, 4
29 la $a0, Titulo #Carrega a mensagem de titulo
30 syscall
31
32 j Solicitar_Notas_Dos_Alunos #Pula para solicitar a 1ª nota
33
34 Nota_Invalida: #Este bloco é usado apenas se ele digitou uma nota invalida
35 li $v0, 4
36 la $a0, Msg_Erro #Mostra a mensagem de erro
37 syscall
38
39 li $v0, 5 #ler um inteiro (nota)
40 syscall
41
42 #Guarda a nota no vetor de notas
43 sw $v0, 0($t2)
44
45 bltz $v0, Nota_Invalida #Se a nota digitada for menor que zero apresenta mensagem de erro
46 bgt $v0, 10, Nota_Invalida #Se a nota digitada for maior que 10 apresenta mensagem de erro
47
48 j Correto # Guarda a nota digita certa agora
49
50 Solicitar_Notas_Dos_Alunos:
51 #Solicita a nota do usuário
52 li $v0, 4
53 la $a0, Msg_Solicitar1 #Carrega a mensagem de inserir nota
54 syscall
55
56 lw $t5, Contador_Alunos #Carrega o contador de alunos para o print
57 addi $t5, $t5, 1 #Incrementa a contagem a cada aluno
58 sw $t5, Contador_Alunos #Guarda a nova contagem
59
60 lw $a0, Contador_Alunos #Carrega a quantidade do aluno atual
61 li $v0, 1 #Imprime o inteiro
62 syscall
63
64 li $v0, 4
65 la $a0, Msg_Solicitar2 #Carrega a mensagem "Aluno"
66 syscall
67
68 li $v0, 5 #ler um inteiro (nota)
69 syscall
70
71 Correto: #linha se ele digitou correto
72 #Guarda a nota no vetor de notas
73 sw $v0, 0($t2)
74
75 bltz $v0, Nota_Invalida #Se a nota digitada for menor que zero apresenta mensagem de erro
76 bgt $v0, 10, Nota_Invalida #Se a nota digitada for maior que 10 apresenta mensagem de erro
77
78
79 #Verifica se a nota é zero se for vai para o bloco de zeros
80 beqz $v0, Notas_Zeradas
81
```

```

82     #Verifica se a nota é maior ou igual a 6 (está aprovado)
83     bge $v0, $t3, Notas_Aprovadas
84
85     #Se não, é reprovado
86     j Notas_Reprovadas
87
88 #Label de alunos aprovados
89 Notas_Aprovadas:
90     lw $t4, Contador_Aprovados    #Carrega o contador e alunos aprovados
91     addi $t4, $t4, 1              #Incrementa a contagem se o aluno foi aprovado
92     sw $t4, Contador_Aprovados    #Guarda a nova contagem
93     j Proximo_Aluno
94
95 #Label de alunos zerados
96 Notas_Zeradas:
97     lw $t4, Contador_Zeros        #Carrega o contador de alunos com nota zero
98     addi $t4, $t4, 1              #Incrementa a contagem se ele tirou 0
99     sw $t4, Contador_Zeros        #Guarda a nova contagem
100
101     #Se ele tirou 0 está reprovado também:
102     lw $t4, Contador_Reprovados    #Carrega o contador de alunos reprovados
103     addi $t4, $t4, 1              #Incrementa a contagem se ele foi reprovado
104     sw $t4, Contador_Reprovados    #Guarda a nova contagem
105     j Proximo_Aluno
106
107 #Label de alunos reprovados
108 Notas_Reprovadas:
109     lw $t4, Contador_Reprovados    #Carrega o contador de alunos reprovados
110     addi $t4, $t4, 1              #Incrementa a contagem se ele foi reprovado
111     sw $t4, Contador_Reprovados    #Guarda a nova contagem
112
113 #Label para passar para o proximo aluno do vetor
114 Proximo_Aluno:
115     addi $t0, $t0, 1              #Incrementa o índice no vetor de notas
116     addi $t2, $t2, 4              #Vai para a próxima posição do vetor de notas adicionando 4 bytes de memoria
117     addi $t1, $t1, -1             #Decrementa o contador de alunos
118
119     bnez $t1, Solicitar_Notas_Dos_Alunos    #Loop para solicitar a nota dos alunos
120
121     #Imprime o vetor de notas
122     li $v0, 4
123     la $a0, Simbolo_Chavel        #Carrega e imprime a mensagem "vetor de notas: ["
124     syscall
125
126     la $t2, Notas_Alunos          #Carrega o endereço de memoria do vetor de notas em $t2
127     li $t1, 10                    #Quantidade total de alunos
128
129 #Imprime as notas do vetor
130 Imprimir_Notas:
131     lw $a0, 0($t2)                #Carrega a nota do vetor em $a0
132     li $v0, 1                     #Imprime um inteiro
133     syscall
134
135     addi $t2, $t2, 4              #Avança para a próxima nota/elemento no vetor
136     addi $t1, $t1, -1             #Decrementa no contador de alunos
137
138     bnez $t1, Imprimir_Virgula    #Verifica se é a última nota se não continua
139
140     #Imprime o vetor de notas
141     li $v0, 4                     #Código ara imprimir string
142     la $a0, Simbolo_Chave2        #Carrega a mensagem em $a0
143     syscall
144
145     j Imprimir_Aprovados          #Volta para imprimir os aprovados
146
147 #Imprime a virgula para separar os elementos do vetor
148 Imprimir_Virgula:
149     li $v0, 4
150     la $a0, Virgula               #Carrega e imprime a virgula
151     syscall
152
153     j Imprimir_Notas
154
155 #Imprime a quantidade de alunos aprovados
156 Imprimir_Aprovados:
157     li $v0, 4
158     la $a0, Msg_Aprovados         #Imprime uma mensagem de alunos aprovados
159     syscall
160
161     lw $a0, Contador_Aprovados    #Carrega a quantidade de alunos aprovados em $a0
162     li $v0, 1                     #Imprime o inteiro

```

```

163 syscall
164
165 #Imprime a quantidade de alunos reprovados
166 li $v0, 4
167 la $a0, Msg_Reprovados #Carrega a mensagem de alunos reprovados
168 syscall
169
170 lw $a0, Contador_Reprovados #Carrega a quantidade de alunos reprovados em $a0
171 li $v0, 1 #Imprime o inteiro
172 syscall
173
174 #Imprime a quantidade de alunos com nota zero
175 li $v0, 4
176 la $a0, Msg_Zerados #Carrega a mensagem de alunos com 0
177 syscall
178
179 lw $a0, Contador_Zeros #Carrega a quantidade de alunos com nota zero em $a0
180 li $v0, 1 #Imprime o inteiro
181 syscall
182
183 #Finaliza o programa
184 li $v0, 10
185 syscall
186

```

## TESTES DE FUNCIONALIDADE

Sistema de Notas:

```

Digite a nota do 1° Aluno: 0
Digite a nota do 2° Aluno: 1
Digite a nota do 3° Aluno: 2
Digite a nota do 4° Aluno: 3
Digite a nota do 5° Aluno: 4
Digite a nota do 6° Aluno: 5
Digite a nota do 7° Aluno: 6
Digite a nota do 8° Aluno: 7
Digite a nota do 9° Aluno: 8
Digite a nota do 10° Aluno: 9
Vetor de notas: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
A quantidade de alunos aprovados: 4
A quantidade de alunos reprovados: 6
A quantidade de alunos com nota igual a zero: 1
-- program is finished running --

```

Sistema de Notas:

```

Digite a nota do 1° Aluno: 0
Digite a nota do 2° Aluno: 0
Digite a nota do 3° Aluno: 0
Digite a nota do 4° Aluno: 0
Digite a nota do 5° Aluno: 0
Digite a nota do 6° Aluno: 0
Digite a nota do 7° Aluno: 0
Digite a nota do 8° Aluno: 0
Digite a nota do 9° Aluno: 0
Digite a nota do 10° Aluno: 0
Vetor de notas: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
A quantidade de alunos aprovados: 0
A quantidade de alunos reprovados: 10
A quantidade de alunos com nota igual a zero: 10
-- program is finished running --

```

```
Sistema de Notas:
Digite a nota do 1° Aluno: 6
Digite a nota do 2° Aluno: 7
Digite a nota do 3° Aluno: 8
Digite a nota do 4° Aluno: 9
Digite a nota do 5° Aluno: 10
Digite a nota do 6° Aluno: 6
Digite a nota do 7° Aluno: 7
Digite a nota do 8° Aluno: 8
Digite a nota do 9° Aluno: 9
Digite a nota do 10° Aluno: 10
Vetor de notas: [6, 7, 8, 9, 10, 6, 7, 8, 9, 10]
A quantidade de alunos aprovados: 10
A quantidade de alunos reprovados: 0
A quantidade de alunos com nota igual a zero: 0
-- program is finished running --
```

```
Sistema de Notas:
Digite a nota do 1° Aluno: 5
Digite a nota do 2° Aluno: 4
Digite a nota do 3° Aluno: 3
Digite a nota do 4° Aluno: 2
Digite a nota do 5° Aluno: 1
Digite a nota do 6° Aluno: 0
Digite a nota do 7° Aluno: 5
Digite a nota do 8° Aluno: 4
Digite a nota do 9° Aluno: 3
Digite a nota do 10° Aluno: 2
Vetor de notas: [5, 4, 3, 2, 1, 0, 5, 4, 3, 2]
A quantidade de alunos aprovados: 0
A quantidade de alunos reprovados: 10
A quantidade de alunos com nota igual a zero: 1
-- program is finished running --
```



```
Sistema de Notas:
Digite a nota do 1º Aluno: -1
Nota invalida, digite uma nota maior que zero e menor que dez: 11
Nota invalida, digite uma nota maior que zero e menor que dez: 5
Digite a nota do 2º Aluno: 5
Digite a nota do 3º Aluno: 6
Digite a nota do 4º Aluno: 7
Digite a nota do 5º Aluno: 8
Digite a nota do 6º Aluno: 9
Digite a nota do 7º Aluno: 0
Digite a nota do 8º Aluno: 4
Digite a nota do 9º Aluno: 15
Nota invalida, digite uma nota maior que zero e menor que dez: 3
Digite a nota do 10º Aluno: 6
Vetor de notas: [5, 5, 6, 7, 8, 9, 0, 4, 3, 6]
A quantidade de alunos aprovados: 5
A quantidade de alunos reprovados: 5
A quantidade de alunos com nota igual a zero: 1
-- program is finished running --
```

### COMENTÁRIOS GERAIS

Para a realização do exercício, a seguinte lógica foi utilizada: De início, o programa solicita que o usuário forneça os dados do aluno, incluindo seu nome e suas notas, isso é feito por um contador "1º Aluno", "2º Aluno". O programa verifica se as notas estão dentro do intervalo válido, entre 0 e 10. Se não, exibe uma mensagem de nota inválida e pede que digite novamente. Após a capturar a nota, o código verifica quais notas são maiores ou iguais a 6 e quais são menores. Ele mantém contadores separados para notas aprovadas e reprovadas. Por fim, o programa imprime o número de alunos aprovados, reprovados e zerados com base nas notas fornecidas. O principal desafio deste exercício foi validar as notas fornecidas pelo usuário no intervalo e manter contadores para alunos aprovados e reprovados, além principalmente da manipulação do vetor, sendo o maior desafio, garantindo que o programa funcione corretamente para diferentes cenários de notas. Foram realizados testes para garantir que o programa funciona conforme o esperado, incluindo casos com notas válidas e inválidas. Todos os testes foram bem-sucedidos, demonstrando que o programa opera conforme as especificações acima.