

ENGENHARIA DE COMPUTAÇÃO

Lab. de Arquitetura de Computadores

1º Projeto

Vítor Ferrara Conti

-

Ra: 19019454

Guilherme Marques Brait Garros

-

Ra: 19264266

1. Introdução

No projeto o grupo aplica a teoria vista em aulas anteriores, a proposta de implementação de um código em MIPS capaz de pequenas interações com alguns dados adicionados pelo usuário, utilizando e reforçando o conhecimento passado pelas aulas teóricas e práticas de Arquitetura de Computadores, assim como a manipulação de vetores, matrizes, operações com valores de ponto flutuante (precisão simples) e endereçamento com pilha.

2. Especificação

2.1 Detalhes de projeto

O Grupo foi capaz de aplicar as funções propostas com êxito, seguindo os objetivos definidos, possibilitando ao usuário executar as seguintes tarefas:

- Cadastrar até 5 alunos seus RAs, atividades e provas.
- Alterar ou adicionar as notas atribuídas aos alunos, diferenciando-os por RA.
- Ordenar a lista de alunos e exibir suas notas.
- Exibir a Média aritmética da Sala e a Relação de Aprovados.

A partir dos Objetivos definidos foi possível estabelecer e dividir algumas porções do projetos, dividindo-os em funções, estabelecemos um total de N funções elas são:

- Main
- CadastroA
- AlterarNotaA
- ExibirNotasT
- ExibirMediaT
- ExibirAprovT
- CalculoDasNotas
- BubbleSortV
- ArredondaNota

Suas respectivas funções e detalhes:

Endereços:

- Matriz em \$a1
- Vetor em \$a3
- Zero float \$f30
- -1 em float \$f31

Main:

A Função main é responsável por receber a matriz e o vetor onde será salvo os dados dos alunos, o vetor de inteiros irá armazenar o RA dos alunos que estarão ligados a matriz que é organizada de forma que as posições "-1" representem uma casa sem valor de nota atribuído.

A Main é responsável também por fazer a coordenação geral para o redirecionamento para as outras funções, com suas respectivas funções ligadas aos objetivos a partir da escolha do usuário feita pela leitura de um inteiro. Ela é executada dentro de um while e portanto o encerramento do programa é feito via interface pelo usuário

CadastroA:

A função Cadastro é responsável por salvar o RA que o usuário cadastrou no Vetor de RAs, fazer a validação da quantidade de RA e de RAs inválidos.

Primeiramente temos um While que verifica se a posição está vazia e outro que verifica se o RA digitado já foi salvo, caso já exista um RA o programa informa uma mensagem de erro e pede uma nova inserção para o usuário, caso tenha atingido o limite de RA o programa informa o usuário e retorna para main

AlterarNotaA:

Essa função é responsável por inserir e alterar a nota do aluno requisitado por RA, primeiro o programa pede ao usuário para inserir o usuário que deseja interagir, a verificação é feita por um for que percorre o vetor de RAs e as linhas da Matriz simultaneamente procurando o RA válido, se não for encontrado, ele mostra uma mensagem de RA inválido e retorna ao menu, se for encontrado ele vai para a função **CalculoDasNotas** e retorna para o programa principal.

ExibirNotaT:

Esta função exibe a nota de todos os alunos, arredonda suas médias, utiliza um BubbleSort para organizar-las por RA e as exibe.

Primeiro a Função chama a **BubbleSortV** para organizar o Vetor e a Matriz simultaneamente, então temos a entrada no FOR de impressão, lá ele anda por todas as posições da matriz na linha do RA até encontrar uma posição vazia ou até o final.

a função **ArredondaNota** é responsável por arredondar as médias parciais antes de printar para o usuário.

ExibirMediaT:

Consiste em exibir a nota média média aritmética final, consiste em somar a nota final de todos os alunos e fazer a média aritmética, dividindo pelo número de alunos cadastrados.

A função anda na matriz e no vetor simultaneamente somando as médias finais de cada aluno e com um contador onde cada ra válido é ++, isso até o final ou até encontrar uma posição vazia, no final as notas somadas são divididas pelo número de alunos, arredondada e exibida.

a função **ArredondaNota** é responsável por arredondar as média final antes de printar para o usuário.

ExibirAprovT:

A Função compara a média final dos alunos com 5 e exibe a taxa de alunos aprovados.

A função anda simultaneamente por um for pelo vetor e pela matriz somando 1 para cada ra existente, comparando se a média final do aluno é ≥ 5 se sim o soma 1 no contador de aprovados e ao final printa na tela aprovados n de n alunos.

CalculoDasNotas:

Função Auxiliar que recebe como parâmetro a posição do aluno no vetor e a posição da nota na matriz, verifica se é Atividade ou Projeto e calcula a média final.

BubbleSortV:

Função Auxiliar Iterativa que visa organizar o Vetor e a Matriz Simultaneamente, utilizando Bubble Sort.

A Função possui um contador que permite limitar em 5x a execução de seu for, ele faz uma verificação $v[i] \leq v[j]$ se sim $i++$, $j++$, se não faz um swap entre eles e $i++$, $j++$.

ArredondaNota:

A Função Auxiliar recebe como parâmetro a nota, pega a parte fracionária da inteira e compara com 0.25 e 0.75, se for menor que 0.25 exibe apenas a inteira, se for maior, compara com 0.75, se menor soma a nota inteira com 0.5 e se for maior soma com 1 e exibe.

2.2 Detalhes de implementação

2.2.1 .data

o .data é usado para armazenar todas suas, strings, variáveis, vetores e matrizes na memória do seu computador

a exetenção

.asciiz é usada para strings

.word para inteiros

.float para precisão simples

para implementar nosso vetor de ra usamos

BancoRa: .word 0, 0, 0, 0, 0 #Vetor RA'S
e a matriz

BancoNotas: .float -1, -1, -1, -1, -1, -1, -1, 0, #matriz precisão simples 5 x 9
-1, -1, -1, -1, -1, -1, -1, 0
-1, -1, -1, -1, -1, -1, -1, 0,
-1, -1, -1, -1, -1, -1, -1, 0,
-1, -1, -1, -1, -1, -1, -1, 0

2.2.2 main

primeiro iniciamos 2 registradores de ponto flutuante, F30 e F31 com valores que seriam utilizados durante o projeto todo
para isso usamos

lwc1 \$f30,Zero #\$f30 vai ser nosso \$fzero
lwc1 \$f31,MenosUm #\$f31 vai ser nosso -1 em float

o Zero e Menosum, foram iniciado no .data e passados para o registrador, vale ressaltar que todos registradores do coprocessador1 iniciam em 0, porém por convenção, ficou determinado carregar esse valor para 1 registrador antes de usar como 0.

depois é printado um menu e o usuário escolhe uma opção. Essa opção passa por beq que serve como "switch case" que determina para qual função o programa deve ir de acordo com a entrada do usuário. Caso a entrada seja inválida o programa pede de novo uma entrada para o usuário.

2.2.3 CadastroA

os desafios de implementação dessa função está nos loop utilizados que servem para andar no vetor.

para selecionar a posição correta do vetor é necessário um loop

que deve ter um i iniciado em 0
add \$t3,\$zero,\$zero

uma label
while:

um somador que anda nas posições do vetor de 4 em 4 semelhante ao i++
addi \$t3,\$t3,4 #passa para próxima posição do vetor

e o branch que ve se ja andou todas as casas que seria necessárias para andar o vetor inteiro. Caso o contrário pula para label
bne \$t3,20,While

nessa função implementados 2 loops, um para andar no vetor preencher e verificar se existem posições vazias e um para verificar se o Ra digitado já existe no vetor

2.2.4 AlterarNotaA

Essa função exigia detalhes de implementação de matrizes. Que exigem um For para l que verifica qual a posição do ra e da sua linha de notas

Fori:

```
move $s1,$zero # i=0;
```

```
addi $t6,$t6,4 # t6 = t6 + 4
```

```
addi $s0,$s0,1 #s0 = s0 +1
```

```
bne $s0,5,ForJ2 # se $s0 != 5 pula para Forl2
```

ao ser validado chama a função de cálculo de notas que será explicada mais a frente

2.2.5 ExibirNotaT

a primeira coisa necessária foi implementar um bubblesort para ordenar os ra's e suas respectivas notas no vetor e na matriz, entretanto separamos um tópico apenas para essa função, mas vale ressaltar que para chama a função dentro da função foi necessário abrir dois espaços, para mandar parâmetros e salvar o endereço de retorno

```
addi $sp,$sp,-8 #abre dois espaços na pilha
```

```
sw $ra,0($sp) #salva ra em sp
```

```
jal BubbleSortV #chamada de função
```

```
lw $ra,0($sp) #devolve o endereço de ra para ra salvo em sp
```

```
addi $sp,$sp,8 #fecha dois espaços na pilha
```

Depois foi necessário implementar 2 for 's i e j, usados para percorrer i - > posição do vetor e linha da matriz e j coluna da matriz

para achar o endereço correto da matriz foi necessário usar o seguinte trecho de código:

```
mul $t1,$s0,8          #i*8
add $t1,$t1,$s1         #i*8+j
sll $t1,$t1,2           #(i*8+j)*4
add $t1,$t1,$a1         #soma endereço da matriz com a posição [i][j]
```

Como os endereço de array em mips andam de 4 em 4 e não de 1 em 1, era necessário, multiplicar o $i * 8$, somar com o j, multiplicar por 4 e depois somar o endereço da matriz

exemplo:

elemento Matriz [1][j] posição em mips = 32

$1 * 8 = 8$

$8 + 0 = 8$

$8 * 4 = 32$

depois é só somar o endereço da matriz

esse comandos são responsáveis por percorrer o vetor e a matriz, permitindo assim imprimir na tela o ra e suas notas

2.2.6 ExibirMediaT

Responsável por calcular a média da turma, seus detalhes são semelhantes aos de funções passadas

vale ressaltar uma otimização, ao comparar o Ra 1 posição do vetor com 0, caso fosse igual, printava média 0 e saia da função, pois ao verificar que a primeira posição é inválida, já sabemos que não existem Ra's no vetor, esse comando impede execução atoa e uma divisão por 0

2.2.7 ExibirAprovT

Responsável por exibir o número de alunos na sala e o número que foi aprovado a função não apresenta implementações novas, mas vale ressaltar o uso do comando auxiliar utilizados diversas vezes ao longo do código

```
li $v0,4 #chamada para printar strings
la $a0,PressEnter #passagem da string a ser escrita
syscall
```

```
move $s7,$a1 #passagem do endereço de a1 para s7
```

```
li $v0,8 #chamada para armazenar strings
```

```
la $a0,buffer #passagem da variavel para salvar string
li $a1,20 #passagem do tamanho da string
syscall
```

```
move $a1,$s7 #passagem do endereço de s7 para a1
```

```
li $v0,4 #chamada para printar strings
la $a0,nextline #passagem do parametro a ser escrito
syscall
```

O código acima permite ao usuário melhor visualização do programa, pois pede que ele digite enter para continuar a execução do programa

2.2.8 CalculoDasNotas

o primeiro item desse código é salvar os valores de s0 e s1 na pilha para poder modificá-los e retornar sem alteração para função original

depois, pega o numero da atividade salvo em \$a2, que se refere ao número da coluna que fica a nota da respectiva atividade e multiplica por 4. Feito isso soma com o valor da linha salvo em t1 também passado como referência da função que a chamou

```
sl $a2,$a2,2 #multiplica a2 por 2
add $t2,$t1,$a2 #soma a posição no endereço do aluno salvo em t1 - Matriz
[$t1][$a2]
```

ao carregar o valor em um registrador, ele era comparado com -1, sinal declarado anteriormente como vazio da posição, para comparar ponto flutuante foi usado os seguintes comandos

```
c.eq.s $f31,$f2 #verifica se são iguais
bc1t altera#se $f2 == f31, pula para valido2
```

caso forem iguais fica salvo 1 em Coprocessador1 e o bc1t, permite pular para alterar, parte do código responsável por atribuir ou alterar a nota. Caso já existisse uma nota na posição, era informado para usuário permitindo escolher entre alterar a nota ou cancelar a operação

caso tudo fosse confirmado, e a entrada do usuário estivesse entre 0 e 10 era atribuída a nota, caso contrário pedia para inserir a nota novamente.

para saber se a nota atribuída era de projeto ou de atividade, o número da atividade, era comparado com 4

```
bgt $a2,4,projeto #se a2 for maior que 4 pula para projeto
se são 5 atividade porque 4? porque a primeira posição do vetor é 0 então a quinta atividade está na posição 4
```

Caso fosse atividade a média era calculada da seguinte forma

$$\text{Média do aluno} = \text{Média do aluno} + (\text{Nota} * 2) / 20$$

caso fosse projeto

$$\text{Média do aluno} = \text{Média do aluno} + (\text{Nota} * 5) / 20$$

2.2.9 BubbleSortV

Responsável por ordenar o vetor é algoritmo iterativo que faz troca de posições do vetor usando

for:

t5 = i

t4 = i + 1

lw \$t6(\$t5) #carrega o valor salvo na posição \$t5 em \$t6

lw \$t8(\$t4) #carrega o valor salvo na posição \$t4 em \$t8

blt \$t6,\$t8,next11#se t6<=t8 pula para next11

sw \$t8,(\$t5) #se não troca seus valores

sw \$t6,(\$t4)

enquanto i+1 <= 5

a lógica de comparação consiste da seguinte forma

v[0] e v[1]

se v[0] <= v[1]

{

 não faz nada

} else

x = v[0]

v[0] = v[1]

v[1] = x

próxima comparação v[1] e v[2]

até v[n-1] com v[n]

esse código é executado pelo número de ras que existentes no vetor.

ou seja n vezes sendo v[n];

2.2.10 BubbleSortM

é uma função auxiliar que caso $v[i] > v[i+1]$ era chamada para trocar a linha das notas

fazendo

$matriz[i][j] = matriz[i+1][j+1]$

e

$matriz[i+1][j+1] = matriz[i][j]$

executado em um for pelo número de colunas

em mips,

`lwc1 $f1,($t1)#carrega valor da matriz em $f1`

`lwc1 $f2,($t2)#carrega valor da matriz em $f2`

`swc1 $f1,($t2) #inverte os valores`

`swc1 $f2,($t1)`

2.2.11 ArredondaNota

o primeiro passo dessa função foi necessário iniciar algumas variáveis

`add.s $f5,$f30,$f1 #f5 = f30 + f1`

\$f5 recebe o valor da média

`lwc1 $f6,zeroVinteCinco #carrega 0.25 em $f6`

`lwc1 $f7,Meio #carrega 0.5 em $f7`

`lwc1 $f8,zeroSetentaCinco #carrega 0.75 em $f8`

`lwc1 $f9,um #carrega 1 em $f9`

depois foi necessário usar os comando `trunc.w.s` e `cvt.s.w` que são responsáveis por pegar apenas a parte inteira do operando

`trunc.w.s $f4,$f1 #pega só a parte inteira de f1 e salva em f4`

`cvt.s.w $f4,$f4`

parte inteira de \$f1 fica salva em \$f4, que vai ser subtraída do \$f5 para podermos lidar apenas com as casas decimais

`sub.s $f5,$f5,$f4 #f5 = f5 - f4`

essa parte decimal é comparada com 0.25 e 0.75

c.lt.s \$f5,\$f6 #compara se f5 é menor que f6
bc1t nota #se for verdade pula pra nota

Caso a parte decimal seja menor do que 0.25, pula para imprimir a nota apenas com a parte inteira

caso contrário compara com 0.75
c.lt.s \$f5,\$f8 #compara se f5 for menor f8
bc1t arredonda # se for verdade pula para arredonda

se for menor que 0.75, arredonda para parte inteira + 0.5
add.s \$f4,\$f4,\$f7 # f4 = f4 + f7

caso contrário,
soma 1 na nota
add.s \$f4,\$f4,\$f9

3. Resultados

3.1 Testes realizados

Bateria de Teste 1: Função cadastro

Para verificar o testes implementamos as seguintes linhas de código no cadastro

```
lw $t9,($t1) #carrega o valor de ra salvo no vetor na posição i  
li $v0,1 #chamada para imprimir inteiros  
move $a0,$t9 #passagem do valor a ser printado  
syscall
```

Teste1:

Cadastro o Ra: 19019454

deveria ficar salvo em \$v0 e de \$v0 passar para o vetor

```
Bem vindo ao controle de notas de Arquitetura de computadores  
[1] Cadastrar aluno  
[2] Adicionar ou alterar nota  
[3] Exibir nota  
[4] Média aritmética das médias da turma  
[5] exibir relação dos aprovados  
[0] encerrar programa  
:1  
  
Entre com o Ra:19019454
```

Ra salvo em \$v0 após armazenar

\$v0	2	19019454
------	---	----------

Print do Ra armazenado no vetor

```
Entre com o Ra:19219454
19219454
```

2 Teste Cadastro de 5 Ra`s :

Nesse teste ocupamos todas as posições do vetor, depois para comprovar que foram realmente armazenados, imprimos usando a função ExibirNotaT. Que deve exibir nesse teste todos os Ra`s sem nenhuma nota
19019454, 19264266, 19012871, 19006568, 19030550.

Resultado:

```
Entre com o Ra:19219454
19219454
```

```
Entre com o Ra:19264266
19264266
```

```
Entre com o Ra:19012871
19012871
```

```
Entre com o Ra:19006568
19006568
```

```
Entre com o Ra:19030550
19030550
```

Usando a função ExibirNotasT para comprovar que foram armazenados

```
Ra:19006568
- Notas:
Media parcial do aluno:0.0
Ra:19012871
- Notas:
Media parcial do aluno:0.0
Ra:19019454
- Notas:
Media parcial do aluno:0.0
Ra:19030550
- Notas:
Media parcial do aluno:0.0
Ra:19264266
- Notas:
Media parcial do aluno:0.0
```

3 teste: mais de 5 ra´s
exibe a seguinte mensagem:

```
Limite de alunos Atingidos
```

4 teste: Ra´s repetidos

ERRO RA EXISTENTE POR FAVOR INSIRA UM DIFERENTE

Bateria de Testes 2: função adicionar ou alterar nota

Teste 1: Vamos adicionar um Ra: 19264266 e adicionar uma nota na atividade 1 e no projeto 2, feito isso vamos imprimir usando a função imprime notas. Que vai imprimir o RA, as notas e parcial.

A Função adiciona nota pede o Ra do aluno e faz uma busca para descobrir em qual linha da matriz deve inserir a nota

Caso o Ra inserido for invalido:

Exibe o erro e volta para main

```
Bem vindo ao controle de notas de Arquitetura de computadores
[1] Cadastrar aluno
[2] Adicionar ou alterar nota
[3] Exibir nota
[4] Média aritmética das médias da turma
[5] exibir relação dos aprovados
[0] encerrar programa
:2

Entre com o Ra:1926

      Erro: Ra Invalido

Pressione [ENTER] para continuar
```

com o Ra correto ele pede qual atividade ou projeto deseja inserir a nota

```
..
Entre com o Ra:19264266

Adicionar ou Alterar Notas
[1] - Atividade 1
[2] - Atividade 2
[3] - Atividade 3
[4] - Atividade 4
[5] - Atividade 5
[6] - Projeto 1
[7] - Projeto 2
[0] - Voltar ao menu principal
:
```

Nota inserida na atividade 1:10

Nota inserida no projeto 2: 8

a Média parcial considera a existencia de todas as atividades e projeto portanto

$\sum_{i=1}^n x_i/20$ No caso descrito $(10 * 2) / 20 + (8 * 5) / 20$, A nota esperada é 3,

```

Bem vindo ao controle de notas de Arquitetura de computadores
[1] Cadastrar aluno
[2] Adicionar ou alterar nota
[3] Exibir nota
[4] Média aritimética das médias da turma
[5] exibir relação dos aprovados
[0] encerrar programa
:3

Ra:19264266
- Notas: 10.0 8.0
Media parcial do aluno:3.0

Pressione [ENTER] para continuar
.

```

Teste 2:

no segundo teste foram inseridos 5 Ra´s e Todas as notas em cada atividade 19019454, 19264266, 19012871, 19006568, 19030550.

```

Ra:19006568
- Notas: 2.0 10.0 8.0 5.5 10.0 7.0 9.5
Media parcial do aluno:7.5
Ra:19012871
- Notas: 8.0 4.5 10.0 5.5 6.0 8.5 8.5
Media parcial do aluno:7.5
Ra:19019454
- Notas: 10.0 2.0 8.5 10.0 7.5 10.0 5.5
Media parcial do aluno:7.5
Ra:19030550
- Notas: 5.0 6.0 7.0 8.0 2.0 4.0 10.0
Media parcial do aluno:6.5
Ra:19264266
- Notas: 10.0 10.0 10.0 10.0 10.0 10.0 10.0
Media parcial do aluno:10.0
Pressione [ENTER] para continuar

```

Teste 3:

adicionamos o ra 192 e a nota de projeto 6 - 10 e a de projeto 7 - 7, depois alteramos a nota do projeto 7 para 8

```

Ra:192
- Notas: 10.0 7.0
Media parcial do aluno:4.0

```

para trocar nota o usuario deve ir na opção 2 do menu principal, digitar o Ra, selecionar a mesma atividade e então a seguinte mensagem vai aparecer

```

Ra:192
- Notas: 10.0 8.0
Media parcial do aluno:4.5

```

```

Esse Aluno Ja tem nota Na atividade 7
[1] - Deseja alterar nota
[2] - cancelar
:

```

Bateria de Testes 3: Função exibir notas

1 Teste: Adicionamos 1 Ra (192) e a nota no projeto 1 - 10 e no projeto 2 -8 o espera do era printar o ra as duas notas e a média 4,5

```

Ra:192
- Notas: 10.0 8.0
Media parcial do aluno:4.5

```

2 Teste: Ideia testa ordenação de Ra's e da matriz

Nesse teste inserimos 5 ra e 10 notas para cada, pois ela deveria ordenar os Ra's e trocar a posição das notas da matriz de acordo com a troca dos ra, ou seja se o ra estava na posição 0 e foi para 1, as notas da linha 0 da matriz deve ir para a linha 1, além disso ela deve arredondar a média do aluno de meio em meio ponto.

Ordem e entrada dos Ra's

19019454, - 10, 2, 8.5, 10, 7.5, 10, 5.5

19264266, - 10, 10, 10, 10, 10, 10, 10

19012871, - 8, 4.5, 10, 5.5, 6, 8.5, 8.5

19006568, - 2, 10, 8, 5.5, 10, 7, 9.5

19030550. - 5, 6, 7, 8, 2, 4, 10

saída do programa

```

Ra:19006568
- Notas: 2.0 10.0 8.0 5.5 10.0 7.0 9.5
Media parcial do aluno:7.5
Ra:19012871
- Notas: 8.0 4.5 10.0 5.5 6.0 8.5 8.5
Media parcial do aluno:7.5
Ra:19019454
- Notas: 10.0 2.0 8.5 10.0 7.5 10.0 5.5
Media parcial do aluno:7.5
Ra:19030550
- Notas: 5.0 6.0 7.0 8.0 2.0 4.0 10.0
Media parcial do aluno:6.5
Ra:19264266
- Notas: 10.0 10.0 10.0 10.0 10.0 10.0 10.0
Media parcial do aluno:10.0

```

Bateria de Teste 4: Teste da função média da turma

Teste 1: apenas 1 aluno

esperado, média do aluno = média da turma

```
Ra:192
- Notas: 10.0 8.0
Media parcial do aluno:4.5
```

```
Média da Sala:4.5
```

teste 2 : 5 Ra's diferentes com notas diferentes

19019454, - 10, 2 , 8.5, 10, 7.5, 10, 5.5

19264266, - 10, 10, 10, 10, 10, 10, 10

19012871, - 8, 4.5, 10, 5.5, 6, 8.5, 8.5

19006568, - 2, 10, 8, 5.5, 10, 7, 9.5

19030550. - 5, 6, 7, 8, 2, 4, 5

saída do programa

```
Ra:19006568
- Notas: 2.0 10.0 8.0 5.5 10.0 7.0 9.5
Media parcial do aluno:7.5
Ra:19012871
- Notas: 8.0 4.5 10.0 5.5 6.0 8.5 8.5
Media parcial do aluno:7.5
Ra:19019454
- Notas: 10.0 2.0 8.5 10.0 7.5 10.0 5.5
Media parcial do aluno:7.5
Ra:19030550
- Notas: 5.0 6.0 7.0 8.0 2.0 4.0 1.0
Media parcial do aluno:4.0
Ra:19264266
- Notas: 10.0 10.0 10.0 10.0 10.0 10.0 10.0
Media parcial do aluno:10.0
```

nesse caso média da sala espera é 7.5

```
Média da Sala:7.5
```

Teste 3: adicionamos mais 1 aluno sem notas

media ta turma = soma das medias / numero de alunos

4,5 / 2 = 2,25

```
Média da Sala:2.5
```

Bateria de testes 5: Teste da função relação de aprovados e Teste de estresse maximo

nesse teste usamos o maximo do programa, cadastramos 5 ra's na seguinte ordem e com as respectivas notas

Ordem e entrada dos Ra's

19019454, - 10, 2 , 8.5, 10, 7.5, 10, 5.5

19264266, - 10, 10, 10, 10, 10, 10, 10

19012871, - 8, 4.5, 10, 5.5, 6, 8.5, 8.5

19006568, - 2, 10, 8, 5.5, 10, 7, 9.5

19030550. - 5, 6, 7, 8, 2, 4, 10

saída do programa

```
Ra:19006568
- Notas: 2.0 10.0 8.0 5.5 10.0 7.0 9.5
Media parcial do aluno:7.5
Ra:19012871
- Notas: 8.0 4.5 10.0 5.5 6.0 8.5 8.5
Media parcial do aluno:7.5
Ra:19019454
- Notas: 10.0 2.0 8.5 10.0 7.5 10.0 5.5
Media parcial do aluno:7.5
Ra:19030550
- Notas: 5.0 6.0 7.0 8.0 2.0 4.0 10.0
Media parcial do aluno:6.5
Ra:19264266
- Notas: 10.0 10.0 10.0 10.0 10.0 10.0 10.0
Media parcial do aluno:10.0
----- (FIM) -----
```

somando todas as media $7,5 + 7,5 + 7,5 + 6,5 + 10 = 39$

$39/5 = 7,8 \rightarrow$ se espera media da sala = 8

```
Média da Sala:8.0
```

Todos foram aprovados portanto deve ter 5 aprovados de 5

```
alunos aprovados:5.0 De Total da Sala:5.0
```

para aprimorar o teste trocamos a nota de projeto dois do ra 19030550 de 10 para 1, assim sua media caiu para 4,0, sendo assim a nova media da sala esperada seria 7,5

```
Ra:19006568
- Notas: 2.0 10.0 8.0 5.5 10.0 7.0 9.5
Media parcial do aluno:7.5
Ra:19012871
- Notas: 8.0 4.5 10.0 5.5 6.0 8.5 8.5
Media parcial do aluno:7.5
Ra:19019454
- Notas: 10.0 2.0 8.5 10.0 7.5 10.0 5.5
Media parcial do aluno:7.5
Ra:19030550
- Notas: 5.0 6.0 7.0 8.0 2.0 4.0 1.0
Media parcial do aluno:4.0
Ra:19264266
- Notas: 10.0 10.0 10.0 10.0 10.0 10.0 10.0
Media parcial do aluno:10.0
```

```
Média da Sala:7.5
```

e o numero de aprovados seria 4 de 5:

:5
alunos aprovados:4.0 De Total da Sala:5.0

3.2 Resultados e discussão

Com o projeto foi possível implementar os conhecimentos a cerca de mips, e aplicar na prática a teoria de uma forma aprofundada. Contribuindo para o crescimento e aprendizado dos integrantes.

O grupo conseguiu alcançar os objetivos e entregar todo o material solicitado, sendo que após sua bateria de teste constatou que o projeto não continha erros e alcançou os resultados esperados, dito que em algumas partes foi necessária, ajuda do professor, monitor e colegas de sala

4. Bibliografia

- <https://www.youtube.com/playlist?list=PL5b07qlmA3P6zUdDf-o97ddfpvPFuNa5A>
- <https://www.embarcados.com.br/serie/mips/>
- Material disponibilizado em aula.
- monitor, professor e colegas de sala
-