

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

INTEGRANTES: Beatriz Morelatto Lorente	RA: 18071597
Christopher de Oliveira Souza	RA: 18726430
Leonardo Sanavio	RA: 18054395
Murilo de Paula Araujo	RA: 17747775

1º PROJETO

1. INTRODUÇÃO

Foi desenvolvido um projeto de gerenciamento de despesas, o qual exerce as funcionalidades de registrar, listar, excluir, exibir gasto mensal, por categoria e exibir ranking de despesas.

Um software como esse pode ser muito útil em ambiente doméstico, pode exibir de forma clara os gastos registrados por mês, e até mesmo exibir de forma decrescente as finanças, mostrando em que mês houve o maior gasto, e por categoria, mostrando quais departamentos estão demandando mais dinheiro. Este programa foi feito na linguagem MIPS, um dos primeiros microprocessadores RISC.

2. ESPECIFICAÇÃO

2.1 DETALHES DE PROJETO

Para armazenar os dados de entrada do teclado foi criada uma struct de 40 bytes, contendo, respectivamente, as informações: ID (4 bytes), Categoria (20 bytes), Dia (4 bytes), Mês (4 bytes), Ano (4 bytes).

As funções são:

Registrar: Faz a leitura dos dados como categoria, dia, mês, ano e armazena na struct e o ID é gerado automaticamente por auto incremento.

Listar: Carrega separadamente todas as informações da struct e exibe na tela.

Excluir: Percorre a struct até encontrar o ID solicitado, quando sua posição é localizada, todos os termos à direita são deslocados uma posição relativa para a esquerda.

Gasto Mensal: Cria um vetor para guardar o gasto de cada mês, e o preenche com os dados do vetor original, verificando o ano e depois o mês, quando o vetor mensal estiver completamente preenchido, será feita a impressão na tela.

Gasto por Categoria: utiliza as ferramentas criadas na função Gasto Mensal.

Ranking: Ordena as despesas, a qual teve o maior gasto para a menor, ...

2.2 DETALHES DE IMPLEMENTAÇÃO

Registrar: Para a leitura dos dados foram utilizados os seguintes registradores, \$t3 para armazenar posição, e \$t0 para o ID, o qual é gerado por auto incremento “addi \$t0, \$t0, 1” e é armazenado na “struct” da seguinte forma “sw \$t0, vetor (\$t3), e para acessar os outros campos como categoria, vai somando a posição como o seguinte exemplo “addi \$t3, \$t3,4”, e para reiniciar guardamos o índice do final do vetor.

Listar: Esta função carrega a posição, assim como os outros campos, da seguinte forma “lw \$a0, vetor (\$t1) ”, depois basta somar o índice para acessar o próximo campo, após percorrer toda a struct, a impressão é feita.

Excluir: Recebe o ID da despesa que deseja excluir com a instrução “li \$v0,5” e percorre o vetor até encontrar, somando o índice (\$t0).

Ao encontrar o ID, todos os campos à direita da despesa a ser excluída são deslocados para a esquerda, e caso o ID não seja encontrado, ou seja, o índice tem um valor maior do que última posição “bgt \$t0, \$t2, alerta” o programa emite uma mensagem de ID não encontrado.

Gasto Mensal: Utiliza os registradores \$t0 e \$t2 para armazenar a primeira e última posição do vetor original, e \$t3 e \$t4 para armazenar o início e o final do vetor mensal, o loop mais externo chamado de loop5 verifica se já foi percorrido todo o vetor original com a instrução “bge \$t0,\$t2” , e se esse for o caso, será feita a impressão, caso contrário, será feita a verificação do vetor mensal “bge \$t3, \$t4”, se o índice deste vetor já está na última posição, será feita uma inserção e depois a verificação do ano e mês.

Gasto por Categoria: Utiliza dos mesmos procedimentos da função Gasto Mensal, só que incrementando a função Bubble Sort para ordenar o vetor em forma crescente, descrevendo e detalhando mais no ranking, onde utilizamos basicamente uma mesma “função” para as duas.

Ranking: A função percorre os campos de “custo” de todas as despesas registradas, em seguida a ordenação é feita pelo método do Bubble Sort, organizando na ordem decrescente os campos de “custos” de cada despesa em um vetor auxiliar. Ou seja, em vez do gasto por categoria que ordenamos o vetor comparando qual elemento é menor, no ranking analisamos o maior, mudando a condição para trocar os seus elementos, salvando o seu endereço de retorno no registrador “ra”, e chamando uma função para realizar a troca desses elementos.

3. RESULTADOS

3.1 TESTES REALIZADOS

Teste 1 com a função Registrar e Imprimir usando valores de ponto flutuante:

```
1 - REGISTRAR DESPESA.
2 - LISTAR DESPESAS.
3 - EXCLUIR DESPESA.
4 - EXIBIR GASTO MENSAL.
5 - EXIBIR GASTO POR CATEGORIA.
6 - EXIBIR RANKING DE DESPESAS.
0 - SAIR.

DIGITE A OPÇÃO DESEJADA: 1

DIGITE A CATEGORIA: teste_um
DIGITE O PREÇO: 1.111111
DIGITE O DIA: 1
DIGITE O MES: 1
DIGITE O ANO: 1

1 - REGISTRAR DESPESA.
2 - LISTAR DESPESAS.
3 - EXCLUIR DESPESA.
4 - EXIBIR GASTO MENSAL.
5 - EXIBIR GASTO POR CATEGORIA.
6 - EXIBIR RANKING DE DESPESAS.
0 - SAIR.

DIGITE A OPÇÃO DESEJADA: 2

CÓDIGO (ID): 1
CATEGORIA: teste_um
VALOR GASTO EM REAIS: 1.111111
DATA: 1/1/1
```

Imagem 1.

Teste 2 com a função Excluir que desloca byte a byte mantendo a formatação dos IDs:

```
CÓDIGO (ID): 1
CATEGORIA: teste_um
VALOR GASTO EM REAIS: 1.111111
DATA: 1/1/1

CÓDIGO (ID): 2
CATEGORIA: teste_um
VALOR GASTO EM REAIS: -1.111111
DATA: 2/2/2

CÓDIGO (ID): 3
CATEGORIA: teste_um
VALOR GASTO EM REAIS: 2.222222
DATA: 3/3/3

CÓDIGO (ID): 4
CATEGORIA: teste_um
VALOR GASTO EM REAIS: -1.111111
DATA: 4/4/4

1 - REGISTRAR DESPESA.
2 - LISTAR DESPESAS.
3 - EXCLUIR DESPESA.
4 - EXIBIR GASTO MENSAL.
5 - EXIBIR GASTO POR CATEGORIA.
6 - EXIBIR RANKING DE DESPESAS.
0 - SAIR.

DIGITE A OPÇÃO DESEJADA: 3

DIGITE O CODIGO DA DESPESA QUE DESEJA EXCLUIR: 2

1 - REGISTRAR DESPESA.
2 - LISTAR DESPESAS.
3 - EXCLUIR DESPESA.
4 - EXIBIR GASTO MENSAL.
5 - EXIBIR GASTO POR CATEGORIA.
6 - EXIBIR RANKING DE DESPESAS.
0 - SAIR.

DIGITE A OPÇÃO DESEJADA: 2

CÓDIGO (ID): 1
CATEGORIA: teste_um
VALOR GASTO EM REAIS: 1.111111
DATA: 1/1/1

CÓDIGO (ID): 3
CATEGORIA: teste_um
VALOR GASTO EM REAIS: 2.222222
DATA: 3/3/3

CÓDIGO (ID): 4
CATEGORIA: teste_um
VALOR GASTO EM REAIS: 1.111111
```

Imagem 2.

Teste 3 com a função Gasto Mensal, que soma as despesas de acordo com o mês:

```
CÓDIGO (ID): 1
CATEGORIA: teste_mensal_um
VALOR GASTO EM REAIS: 200.5
DATA: 1/1/1

CÓDIGO (ID): 2
CATEGORIA: teste_mesnal_doi
VALOR GASTO EM REAIS: 50.5
DATA: 1/4/1

CÓDIGO (ID): 3
CATEGORIA: teste_mensal_um
VALOR GASTO EM REAIS: -30.25
DATA: 5/1/1

1 - REGISTRAR DESPESA.
2 - LISTAR DESPESAS.
3 - EXCLUIR DESPESA.
4 - EXIBIR GASTO MENSAL.
5 - EXIBIR GASTO POR CATEGORIA.
6 - EXIBIR RANKING DE DESPESAS.
0 - SAIR.

DIGITE A OPÇÃO DESEJADA: 4
VALOR GASTO EM REAIS: 170.25
MÊS/ANO:1/1
VALOR GASTO EM REAIS: 50.5
MÊS/ANO:4/1
```

Imagem 3.

Teste 4 com a função Exibir Gasto por Categoria que faz a soma dos valores de mesma categoria:

Processo executado: $1.111111 + 2.222222 - 1.111111 = 2.222219$

```
1 - REGISTRAR DESPESA.
2 - LISTAR DESPESAS.
3 - EXCLUIR DESPESA.
4 - EXIBIR GASTO MENSAL.
5 - EXIBIR GASTO POR CATEGORIA.
6 - EXIBIR RANKING DE DESPESAS.
0 - SAIR.

DIGITE A OPÇÃO DESEJADA: 5

      CATEGORIA      PRECO
1  Teste um          2.222219
```

Imagem 4.

Teste 5 com a função Ranking de Despesas:

Despesas listadas (Imagem 5).

DIGITE A OPÇÃO DESEJADA: 2

CÓDIGO (ID): 1

CATEGORIA: t_rank_um

VALOR GASTO EM REAIS: 500.5

DATA: 1/1/1

CÓDIGO (ID): 2

CATEGORIA: t_rank_dois

VALOR GASTO EM REAIS: 100.1

DATA: 2/2/2

CÓDIGO (ID): 3

CATEGORIA: t_rank_tres

VALOR GASTO EM REAIS: 250.25

DATA: 3/3/3

Imagem 5.

Impressão do Ranking, ele ordena do maior gasto para o menor. (Imagem 6).

```
1 - REGISTRAR DESPESA.  
2 - LISTAR DESPESAS.  
3 - EXCLUIR DESPESA.  
4 - EXIBIR GASTO MENSAL.  
5 - EXIBIR GASTO POR CATEGORIA.  
6 - EXIBIR RANKING DE DESPESAS.  
0 - SAIR.
```

DIGITE A OPÇÃO DESEJADA: 6

	CATEGORIA	PRECO
1	T rank um	500.5
2	T rank tres	250.25
3	T rank dois	100.1

Imagem 6.

3.2 RESULTADOS E DISCUSSÃO

No trabalho realizado, o grupo conseguiu finalizar o programa em pouco tempo, dividindo as tarefas e partes do código como preferido. Os integrantes se reuniram nos primeiros dias para decidir como seria feita a estrutura dos dados do programa (armazenamento, busca de informações, etc). Houve certa dificuldade na realização da proposta 5, e o grupo levou um tempo maior na solução da opção requerida.

Com esse trabalho, as habilidades de programação na linguagem Assembly MIPS foram aprimoradas, e o grupo obteve mais experiência na resolução de problemas abordados, e que foram surgindo no decorrer do programa. Os membros se reuniam nos laboratórios e também virtualmente em suas residências para realizar respectivas partes do programa e sanar dúvidas surgidas no código. Depois de vários testes realizados, e muitos erros corrigidos, o programa finalmente foi concluído.