



UFC

TRABALHO FINAL DE PROGRAMAÇÃO COMPUTACIONAL

Sistema Estatístico de Rendimento Estudantil (SERE)

ÍNDICE

1. Origem;
2. Objetivo Geral;
3. Instruções de Uso;
4. Dificuldades;
5. Considerações Finais;
6. Referências.

ORIGEM

- Observando as dificuldades encontradas pelos professores no decorrer da vida acadêmica, percebemos a necessidade de um programa capaz de avaliar e retornar em números precisos o desempenho dos alunos.
- Uma ferramenta com um objetivo inicial similar ao SERE é o ENEM.
- Público-alvo: professores.
- Software focado no pós-prova.

OBJETIVO GERAL

O presente projeto visa suprir a carência identificada por meio de algoritmos em linguagem C, além de automatizar um processo bastante repetitivo e cansativo pós-correção de provas.

INSTRUÇÕES DE USO: Cadastro das Questões.

Primeiro passo:

```
A SEGUIR O PROGRAMA DE AVALIAÇÃO DE TURMA
```

```
LEMBRETE:
```

- 1 - Leia primeira a documentação do projeto para utilizar.
- 2 - Você tem a limitação de escolher no máximo 10 questões.

```
Informe quantas questões terá na sua prova: 5_
```

Segundo passo:

```
Assuntos do primeiro semestre de programação:
```

- 0. Histórico da Computação
- 1. Arquitetura Básica
- 2. Sistemas de Numeração
- 3. Lógica de Programação
- 4. Pseudocódigo
- 5. Fluxogramas
- 6. Sintaxe da Linguagem C
- 7. Variáveis
- 8. Funções de Entrada e Saída
- 9. Estruturas Condicionais em C
- 10. Comandos de Repetição em C
- 11. Comandos de Controle de Fluxo
- 12. Vetores
- 13. Matrizes
- 14. Strings
- 15. Structs e Typedef
- 16. Ponteiros
- 17. Malloc / Calloc / Free
- 18. Funções
- 19. Arquivos

```
ATENÇÃO: Informe a numeração referente a 1ª questão: _
```

INSTRUÇÕES DE USO: Cadastro das Questões.

Terceiro passo:

Digite:

1 se a questão possuir nível fácil;
2 se a questão possuir nível médio;
3 se a questão possuir nível difícil.

Selecione a dificuldade da 1ª questão sobre Fluxogramas: 1

Selecione a dificuldade da 2ª questão sobre Arquivos: 3

Selecione a dificuldade da 3ª questão sobre Variáveis: 1

Selecione a dificuldade da 4ª questão sobre Fluxogramas: 2

Selecione a dificuldade da 5ª questão sobre Arquitetura Básica: 1

INSTRUÇÕES DE USO: Cadastro dos Alunos.

Quarto passo:

```
Informe a quantidade de alunos que realizaram a prova: 5
```

Quinto passo:

```
Informe o nome do aluno número 1: ana julia  
Informe o nome do aluno número 2: pedro  
Informe o nome do aluno número 3: carla  
Informe o nome do aluno número 4: joao  
Informe o nome do aluno número 5: maria
```

INSTRUÇÕES DE USO: Cadastro dos Alunos.

Sexto passo:

Digite:

1 se o aluno acertou a questão;

0 se o aluno errou a questão.

ATENÇÃO: Informe o Gabarito de ana julia.

Informe se ana julia acertou ou errou a 1ª questão sobre Fluxogramas: 1

Informe se ana julia acertou ou errou a 2ª questão sobre Arquivos: 0

Informe se ana julia acertou ou errou a 3ª questão sobre Variáveis: 1

Informe se ana julia acertou ou errou a 4ª questão sobre Fluxogramas: 1

Informe se ana julia acertou ou errou a 5ª questão sobre Arquitetura Básica: 1

INSTRUÇÕES DE USO: Desempenho da Turma.

Acertos e Erros (em porcentagem)

ANÁLISE GERAL DA TURMA: PERCENTUAL DE ACERTOS EM CADA QUESTÃO

QUESTÃO 1 - Fluxogramas

Percentual de Acertos/Erros: 60,00% / 40,00%.

QUESTÃO 2 - Arquivos

Percentual de Acertos/Erros: 60,00% / 40,00%.

QUESTÃO 3 - Variáveis

Percentual de Acertos/Erros: 80,00% / 20,00%.

QUESTÃO 4 - Fluxogramas

Percentual de Acertos/Erros: 100,00% / 0,00%.

QUESTÃO 5 - Arquitetura Básica

Percentual de Acertos/Erros: 80,00% / 20,00%.

Análise (por nível de dificuldade da questão)

ANÁLISE GERAL DA TURMA: CONDIÇÃO DA TURMA

QUESTÃO 1 - Fluxogramas => FÁCIL

Desempenho da Turma: BOM

QUESTÃO 2 - Arquivos => DIFÍCIL

Desempenho da Turma: ÓTIMO

QUESTÃO 3 - Variáveis => FÁCIL

Desempenho da Turma: ÓTIMO

QUESTÃO 4 - Fluxogramas => MÉDIO

Desempenho da Turma: ÓTIMO

QUESTÃO 5 - Arquitetura Básica => FÁCIL

Desempenho da Turma: ÓTIMO

INSTRUÇÕES DE USO: Análise de Desempenho.

- Questão com Nível Fácil:

- Ótimo: Desemp. $\geq 75\%$
- Bom: $60\% \leq \text{Desemp.} < 75\%$
- Regular: $45\% \leq \text{Desemp.} < 60\%$
- Ruim: $30\% \leq \text{Desemp.} < 45\%$
- Crítico: Desemp. $< 30\%$

- Questão com Nível Médio:

- Ótimo: Desemp. $\geq 60\%$
- Bom: $45\% \leq \text{Desemp.} < 60\%$
- Regular: $35\% \leq \text{Desemp.} < 45\%$
- Ruim: $20\% \leq \text{Desemp.} < 35\%$
- Crítico: Desemp. $< 20\%$

- Questão com Nível Difícil:

- Ótimo: Desemp. $\geq 45\%$
- Bom: $35\% \leq \text{Desemp.} < 45\%$
- Regular: $20\% \leq \text{Desemp.} < 35\%$
- Ruim: $15\% \leq \text{Desemp.} < 20\%$
- Crítico: Desemp. $< 15\%$

DIFICULDADES: Falta de conhecimento sobre Struct.

Algoritmo de definir dificuldade de cada questão sem Struct comparada após a utilização de Struct.

```
for(){
    if(){
        for(cont = 0; cont <= 19; cont++){
            if(cont=questselecionadas[i]){
                printf("Como você classifica o nível de dificuldade da questão sobre %s da sua prova ?");
                scanf("d", dificuldade[i]);
                printf("\n");
                system("pause");
                system("cls");
            }
        }
    }
}
```

```
for(i = 0; i < quantquest; i++){
    do{
        printf("Selecione a dificuldade da %dª questão sobre %s: ", i + 1, banco[i].tema);
        scanf("%d", &bancoNivel[i].nivel);
        if(bancoNivel[i].nivel<1 || bancoNivel[i].nivel>3)
            printf("\nDIGITE UM VALOR VÁLIDO.\n\n");
    }while(bancoNivel[i].nivel<1 || bancoNivel[i].nivel>3);
    printf("\n");
}
```

DIFICULDADES: Utilizando Scanf com String.

Como armazenar o nome completo de um aluno em uma string e fechar o ciclo do scanf somente quando o usuário apertar ENTER.

```
//Informar dados de cada aluno

for(i = 0; i < qalunos; i++){
    dados[i].indice = i+1;

    printf("Informe o nome do aluno número %d: ", dados[i].indice);

    scanf(" %[^\\n]s", dados[i].nome);
}
```

DIFICULDADES: Utilizando Structs.

Passar as questões armazenadas em um vetor de caracteres para um outro vetor de estrutura que contém uma string.

```
char questoes[20][100] = {"Histórico da Computação", "Arquitetura Básica", "Sistemas de Numeração",  
                           "Lógica de Programação", "Pseudocódigo", "Fluxogramas", "Sintaxe da Linguagem C",  
                           "Variáveis", "Funções de Entrada e Saída", "Estruturas Condicionais em C",  
                           "Comandos de Repetição em C", "Comandos de Controle de Fluxo",  
                           "Vetores", "Matrizes", "Strings", "Structs e Typedef", "Ponteiros",  
                           "Malloc / Calloc / Free", "Funções", "Arquivos"};
```

```
struct inbox{  
    int questsel;  
    char questselLetra[100];  
};  
  
struct inbox questdificuldade[10];
```

```
for(i = 0; i < quantquest ; i++){  
    int aux = questselecionadas[i];  
  
    //armazena a numeração das questões  
    questdificuldade[i].questsel = aux;  
  
    //armazena a questão digitada  
    strcpy(questdificuldade[i].questselLetra, questoes[aux]);  
}
```

DIFICULDADES: Variáveis em Excesso.

Houve confusão com a alta quantidade de variáveis (assim como estruturas e vetores), dificultando a fluidez do desenvolvimento. Vale destacar que algumas variáveis não possuíam um nome relacionado com sua função no código, situação também aplicável a Arrays e Structs do código.

```
for(i = 0; i < quantquest; i++){
    varaux1 = 0;
    for(j = 0; j < qalunos; j++){
        varaux1 += dados[j].notas[i];
    }
    varaux1 = varaux1 / (float) qalunos;
    varaux1 *= 100;
    varaux2 = 100 - varaux1;
    printf("QUESTÃO %d - %s\nPercentual de Acertos/Erros: %.2f%% / %.2f%%.\n\n", i + 1, questdificuldade[i].questselLetra, varaux1, varaux2);
}
```

DIFICULDADES: Excesso de Código na Função Principal.

Com o excesso de linhas de código na função main, ficou mais difícil encontrar erros específicos, assim como dificultou bastante a leitura e entendimento do código como um todo. Dessa maneira foi necessária a utilização de funções.

```
34 int main(void){  
82  
83 void cadQuestoes(int quantquest){  
151  
152 void cadAluno(Aluno dados[],int qalunos,int quantquest){  
193  
194 void acertoErro(Aluno dados[], int quantquest,int qalunos){  
213  
214 void condicaoTurma(Aluno dados[], int quantquest,int qalunos){  
283  
284
```

CONSIDERAÇÕES FINAIS

A partir do projeto apresentado, foi possível desenvolver competências de acentuada importância para o mercado de trabalho, como o trabalho em equipe e, de maneira evidente, as habilidades em programação, sendo, esta última, uma habilidade indispensável de um profissional da tecnologia da informação que visa uma carreira promissora.

REFERÊNCIAS

ENEM - APRESENTAÇÃO. Disponível em:

<http://portal.mec.gov.br/enem-sp-2094708791#:~:text=Criado%20em%201998%2C%20o%20Exame,ensino%20m%C3%A9dio%20em%20anos%20anteriores>. Acessado em 14/06/2022.

OBRIGADO PELA ATENÇÃO