



O futuro com inovação

## **Projeto**

APROVA+

### **GRUPO 4**

Geziel de Andrade - 43608906

João Vitor Galeano Ferraz – 43799523

Pedro Viana da Silva - 43477089

Vinicius Eduardo Santos de Oliveira - 41799305

Este documento reúne a documentação, diário de bordo, código e apresentação do projeto desenvolvido pela CSYSTEM.

maio de 2025

## APROVA+

### CSYSTEM

**Cliente:** Sra. Renata Costa

**Centro Universitário:** Braz Cubas

**Data de entrega:** 27 de maio de 2025

### OBJETIVO DO PROJETO:

Desenvolver um sistema de cálculo de notas em linguagem C para o Centro Universitário Braz Cubas. O sistema APROVA+ será capaz de calcular a média das notas dos alunos e apresentar os resultados de forma interativa e eficiente. O projeto usará conceitos de programação básica como estruturas de controle, laços de repetição, vetores, structs, e outros conceitos relevantes pesquisados e durante as aulas.

### PEDIDO:

A Universidade Braz Cubas solicitou o desenvolvimento de um sistema em linguagem C, para realizar o cálculo da média dos alunos com base em duas notas principais:

- A1: Avaliação aplicada pela instituição
- A2: Avaliação aplicada pelo professor

Caso o aluno não alcance a média mínima de 6,0, ele terá direito a uma Avaliação Final (AF), que substituirá a menor das duas notas (A1 ou A2). Todas as avaliações possuem valor de 0 a 5. O sistema deverá processar os dados e informar se o aluno está aprovado ou reprovado, de acordo com a média final.

### REQUISITOS DO SISTEMA:

#### Requisitos Funcionais:

**Cadastro de notas:** O sistema deve permitir a inserção das notas A1 e A2. Caso a nota inserida esteja fora do intervalo de 0 a 5, o sistema deve solicitar um novo valor até que uma entrada válida seja fornecida.

**Cálculo de média:** O sistema deve somar as notas A1 e A2, se a soma for maior ou igual a 6, o aluno é aprovado, senão, a soma for menor que 6, o aluno

precisará fazer a Avaliação Final (AF). A nota da AF deve substituir a menor nota entre A1 e A2, após a AF, a nova média deve ser recalculada para determinar a aprovação ou reprovação.

**Exibições de resultados:** O sistema deve informar se o aluno está aprovado ou se precisará fazer a AF, depois da AF, o sistema deve exibir se o aluno está aprovado ou reprovado.

### **Requisitos não funcionais**

- O sistema será desenvolvido com suporte a estruturas como while, switch/case, if/else, do/while, for, struct, CRUD.
- O processamento das informações será feito de maneira eficiente, possivelmente utilizando dicionários.
- A interface será simples e intuitiva para facilitar o uso pelos alunos.

### **Requisitos do Usuário**

O sistema deve ser de fácil acesso para os professores, devendo ter a possibilidade de revisar as notas antes de finalizar o cálculo.

### **Requisitos do Sistema**

- O sistema deve ser responsivo e rodar em navegadores modernos ou como um aplicativo desktop leve.
- Deve validar as entradas dos usuários e fornecer feedback imediato.

### **Requisitos do Produto**

O sistema deve garantir um ambiente seguro para armazenar temporariamente as notas durante a sessão do usuário.

### **Requisitos do Domínio**

O sistema deve seguir as diretrizes acadêmicas estabelecidas no manual do aluno, as notas devem obrigatoriamente estar no intervalo de 0 a 5.

### **Requisitos Organizacionais**

- O sistema deve estar alinhado com as normas de avaliação da instituição.

- O uso deve ser restrito à comunidade acadêmica da faculdade.

### **Requisitos Externos**

O sistema pode necessitar de acesso ao manual do aluno para atualizações futuras das regras de cálculo de média.

### **SISTEMA:**

#### **Objetivos de Negócio:**

O APROVA+ é sistema desenvolvido oferece uma solução prática para o controle das notas dos alunos, facilitando o trabalho da cliente, que geralmente é a professora responsável pela disciplina. Com ele, é possível cadastrar, alterar, excluir e consultar as informações de cada aluno de forma rápida e segura, eliminando a necessidade de registros manuais que são mais suscetíveis a erros. A atualização das notas é feita com base no nome e no RGM do aluno, garantindo maior precisão e evitando duplicidades.

Para a faculdade, o sistema representa uma modernização na gestão acadêmica, automatizando processos que antes eram realizados manualmente. Isso contribui para a padronização dos dados e para a redução de erros administrativos. Além disso, o sistema facilita o acompanhamento do desempenho dos alunos, permitindo que a instituição tenha um controle mais eficiente e confiável sobre as informações acadêmicas.

Com essa ferramenta, a cliente ganha em praticidade e agilidade, podendo dedicar mais tempo ao ensino. A faculdade, por sua vez, fortalece sua organização interna e demonstra compromisso com o uso de tecnologia para melhorar seus processos educacionais.

#### **Funcionalidades e Requisitos Técnicos:**

O APROVA+ deve permitir que o usuário insira as notas dos alunos de forma prática e eficiente, garantindo a correta captura das informações. Além disso, o sistema deve calcular automaticamente as médias das notas, fornecendo os resultados de maneira clara e acessível para facilitar a interpretação do desempenho dos alunos.

É imprescindível que o sistema atenda ao modelo CRUD, possibilitando as operações de criação, leitura, atualização e exclusão dos registros dos alunos. Essas funcionalidades são essenciais para a gestão completa e dinâmica dos

dados, assegurando que as informações estejam sempre atualizadas e organizadas.

Para garantir o funcionamento adequado, o sistema deve assegurar a integridade dos dados durante as operações, evitar duplicidades e oferecer uma interface simples para interação do usuário, promovendo uma experiência eficiente e sem erros.


### **Tecnologias Utilizadas:**

APROVA+ foi desenvolvido utilizando a linguagem C, por ser uma linguagem estruturada e amplamente usada no ensino de programação. Ela permite compreender melhor a lógica dos algoritmos e o funcionamento interno de um programa. Para organizar os dados dos alunos, como nome, RGM e nota, foram utilizadas estruturas chamadas structs, que facilitam o agrupamento de informações relacionadas.

O controle do fluxo do sistema foi realizado com estruturas de decisão, como if e switch/case, além de laços de repetição, como for, while e do/while, permitindo interações com o usuário e repetição de tarefas conforme necessário. Também foram utilizadas funções de entrada e saída, como scanf e printf, para receber e exibir informações.

Além disso, o projeto utilizou bibliotecas padrão como stdio.h, string.h e locale.h. Esta última foi usada para configurar a linguagem do sistema, permitindo o uso correto de acentuação e caracteres especiais. Com essas ferramentas, o sistema conseguiu realizar as operações de cadastro, consulta, atualização e exclusão de alunos, seguindo a lógica CRUD de forma clara e eficiente.

### **CRONOGRAMA:**

 CRONOGRAMA - APROVA+				
Fase	Descrição	Início	Término	Atualização
Planejamento	Definição do escopo e cronograma	01/abr	02/abr	08/abr
Requisitos	Identificação das funcionalidades necessárias	02/abr	04/abr	-
Projeto do Sistema	Estruturação do sistema e definição de fluxos	29/abr	13/mai	Constante
Codificação	Desenvolvimento do código-fonte em C	06/abr	26/mai	-
Testes	Verificação de funcionamento e correções	20/mai	26/mai	27/mai
Documentação	Criação da documentação técnica e do projeto	20/mai	24/mai	25/mai
Entrega Final	Apresentação e entrega do sistema completo	27/mai		

## **Marcos Importantes:**

### **Adição do CRUD (Create, Read, Update, Delete)**

Implementação das funções principais de cadastro, listagem, atualização e exclusão de registros. Esse marco representa a entrega da base funcional do sistema, garantindo que os dados dos alunos possam ser manipulados com flexibilidade.

### **Adição da Struct para Aluno**

Criação da estrutura struct, permitindo organizar de forma eficiente os dados (nome, RGM, nota). Esse marco tornou o código mais limpo, modular e preparado para futuras expansões.

### **Criação do Menu com Switch/Case**

Estruturação de um menu interativo que usa switch/case para que o usuário possa escolher facilmente a operação desejada. Esse marco garante usabilidade e clareza na navegação do sistema.

## APROVA+

### **Vinicius Eduardo**

Gerente de Projeto: Responsável por organizar as tarefas e coordenar a equipe.

Desenvolvedores: Divisão do código entre os membros para a implementação das funcionalidades.

Qualidade e Documentação: Realização de testes no sistema e elaboração de documentação empresarial.

### **João Vitor**

Designer e Marketing: Responsável pela identidade visual da empresa, comunicação e apoio na documentação.

### **Geziel de Andrade**

Documentação: Elaboração do Português Estruturado e apoio.

### **Pedro Viana da Silva**

Designer e Marketing: Responsável comunicação da empresa e apoio no código

## **CSYSTEM:**

### **Apresentação da Empresa:**

A CSYSTEM foi fundada em 2025 por um grupo de jovens apaixonados por tecnologia e inovação, com o objetivo de criar softwares que facilitassem a vida das pessoas e impulsionassem o crescimento das empresas. Desde sua criação, a CSYSTEM tem se dedicado a desenvolver soluções eficientes, confiáveis e personalizadas, sempre acompanhando as tendências do mercado tecnológico.

### **Missão:**

Desenvolver soluções de software inovadoras e de alta qualidade, que atendam às necessidades específicas dos clientes, promovendo a transformação digital e o crescimento sustentável.

### **Visão:**

Ser reconhecida como uma empresa referência em tecnologia, inovação e atendimento ao cliente, contribuindo para a evolução do setor de desenvolvimento de software no Brasil e internacionalmente.

### **Valores:**

- Compromisso com a qualidade e excelência.

- Inovação contínua.
- Transparência e ética nas relações.
- Foco no cliente e suas necessidades.
- Trabalho em equipe e valorização das pessoas.

### **Objetivo do Projeto:**

O projeto APROVA+, desenvolvido pela CSYSTEM será entregue com a funcionalidade de cadastrar alunos, registrar suas notas, calcular automaticamente a média e exibir se estão aprovados ou reprovados, de forma clara e organizada. Além disso, permitirá ao usuário consultar, atualizar ou excluir registros, seguindo a lógica CRUD.

Com uma interface simples e interativa via terminal, o sistema garante facilidade no uso e precisão nos resultados, atendendo às necessidades de organização e controle acadêmico. A entrega inclui um código-fonte funcional, documentado e pronto para ser executado em qualquer ambiente compatível com a linguagem C.

### **Metodologia de Trabalho:**

A CSYSTEM utilizou uma metodologia ágil no desenvolvimento do sistema, com foco em organização, flexibilidade e entregas contínuas. O processo foi dividido em etapas curtas, nas quais as funcionalidades principais, como cadastro de alunos, cálculo de médias e operações CRUD, foram sendo implementadas e testadas individualmente. Isso permitiu identificar e corrigir possíveis falhas ainda durante o desenvolvimento.

A cada etapa concluída, o sistema era validado para garantir o bom funcionamento e atender aos requisitos definidos. Essa abordagem facilitou a comunicação entre os envolvidos e contribuiu para uma entrega mais eficiente, com um sistema funcional, organizado e alinhado às necessidades do cliente.

### **Versão 1.0**

A primeira versão do código tem como objetivo avaliar o desempenho de um aluno a partir de suas notas e determinar se ele está aprovado ou não. O



programa solicita duas notas iniciais, chamadas de A1 e A2. Com essas duas notas, o código calcula a média aritmética. Caso essa média seja maior ou igual a 6, o aluno é considerado aprovado e o programa encerra com essa informação.

No entanto, se a média for inferior a 6, o aluno terá a oportunidade de realizar uma Avaliação Final (AF). Nesse caso, o código solicita a nota da AF e então substitui a menor nota entre A1 e A2 por essa nova nota. Com a nova combinação de notas, a média é recalculada. Se a nova média for maior ou igual a 6, o aluno é aprovado com base na nota da AF; caso contrário, o aluno é considerado reprovado.

### **Versão 2.0**

Na nova versão do sistema, foi pedido que, além de calcular as notas de um aluno, ele também fosse capaz de fazer o mesmo processo para 100 alunos. Além disso, foi solicitado que fossem adicionadas as informações do nome e do RGM (registro do aluno) para cada um deles, para facilitar a identificação individual no resultado final.

Todo o restante do funcionamento do código permaneceu como na versão anterior: o sistema continua pedindo duas notas, calcula a média e, caso essa média seja menor que 6, solicita a nota da avaliação final, substituindo a menor das duas anteriores para recalculer a média. Ao final, além de informar se cada aluno foi aprovado ou reprovado, foi adicionado também o cálculo da média geral da turma, com base nas médias finais dos 100 alunos.

### **Versão Final**

Na versão final do sistema, foram adicionadas novas funcionalidades para torná-lo mais completo e prático para o uso do professor. A principal novidade foi a inclusão de um menu com opções, permitindo ao professor escolher o que deseja fazer dentro do sistema, como cadastrar um novo aluno, consultar informações, alterar dados, excluir registros ou visualizar os resultados. Com isso, o sistema passou a ter um CRUD completo — ou seja, permite Cadastrar, Consultar, Atualizar e Deletar os dados dos alunos.

Também foi incluído um relatório final individual para cada aluno, que mostra todas as informações registradas, como nome, RGM, notas, média e situação

(aprovado ou reprovado). Todo o restante da lógica do código continua igual: o cálculo das médias, a substituição pela nota da avaliação final quando necessário e o cálculo da média geral da turma ao final do processo.

### **Diferenciação e Inovação:**

APROVA+ se destaca pela simplicidade e eficiência na gestão de notas escolares, oferecendo uma solução prática e funcional para controle acadêmico. Sua estrutura em linguagem C torna o sistema leve, de fácil execução e com rápida resposta, ideal para ambientes educacionais com recursos limitados.

A proposta de valor está na automação das tarefas manuais, como cálculo de médias e organização de dados, reduzindo erros e otimizando o tempo do usuário. Além disso, a implementação completa do modelo CRUD garante controle total sobre as informações dos alunos, promovendo clareza, organização e confiabilidade nos resultados.

### **Solução Técnica e Comercial:**

O sistema APROVA+ foi desenvolvido com foco em atender de forma direta e eficiente às necessidades do cliente, oferecendo uma solução simples, funcional e acessível para o controle de notas escolares. Utilizando a linguagem C, foram aplicadas estruturas de controle, laços de repetição e structs para organizar e processar os dados dos alunos com precisão.

Com base no modelo CRUD, o sistema permite cadastrar, consultar, atualizar e excluir registros, garantindo total controle sobre as informações acadêmicas. Comercialmente, a APROVA+ trata-se de uma solução de fácil implementação, baixo custo e que não depende de softwares complexos, sendo ideal para instituições que buscam praticidade e organização no acompanhamento do desempenho dos alunos.

## **Diário de Bordo**

### **Dia 1 – Reunião Inicial com o cliente (01/04/2025)**

A cliente fez o pedido de um sistema que se calcula as notas de um aluno, sendo elas, “Avaliação 1 (A1), Avaliação 2 (A2) e Avaliação Final (AF)”. Tudo isso em linguagem C, utilizando “if/else, while e Switch/case”.

Após toda a explicação os integrantes entraram no grupo da BlackBoard e começaram o Brainstorming (Chuva de ideias). Uma delas foi o nome da empresa, que por unanimidade foi escolhido o nome “CSystem”, onde o “C” representa a linguagem C e a Computação, já o “System” de sistema. Com isso, criamos a nossa logo e slogan.



# **CSystem**

O futuro com inovação

Desta forma, surge a empresa e o projeto. Contudo, como somos uma empresa profissional na área de T.I, o nosso projeto tem o nome de “APROVA+”, significando que o nosso objetivo além de entregar o nosso sistema de maneira eficiente e com qualidade, pensamos nos alunos e algo que chamasse a atenção deles e dos professores, com o intuito, de fazer os alunos buscarem pela aprovação cada vez mais.

### **Versão 0.1**

Na primeira reunião com o cliente sobre o projeto, a versão calcularia a nota de um aluno, tendo a A1, A2 e AF se necessário. Todas as notas deveriam de 0 a 5 e somando as duas teriam que ser maior que 6 para que o aluno fosse aprovado, caso contrário, o aluno precisaria fazer a avaliação final que substituiria a menor nota entre as duas.

Para isso foi necessário fazer um planejamento de como seria o andamento do projeto, no primeiro momento seria necessário realizado os requisitos do sistema e o português estruturado.

## **Dia 2 – Segunda reunião com o cliente (23/04/2025)**

Foram realizadas algumas pesquisas sobre as funções do **CRUD** e como usar, também ocorreu alterações pelo cliente, onde foi pedido: que o usuário consiga cadastrar não só as notas, mas sim o nome dos alunos, RGM dos alunos e o relatório final mostrando todos os dados utilizados.

### **Versão 0.2**

Na nova versão do sistema, foi pedido que, além de calcular as notas de um aluno, ele também fosse capaz de fazer o mesmo processo para 100 alunos. Além disso, foi solicitado que fossem adicionadas as informações do nome e do RGM (registro do aluno) para cada um deles, para facilitar a identificação individual no resultado final.

## **Dia 3 – Terceira e última reunião com o cliente (**

Na última reunião com a cliente sobre o projeto, ela pediu um menu para que os professores pudessem escolher o que fazer, sendo necessário (cadastrar, atualizar, listar e excluir) as notas dos alunos ou os alunos em si.

### **Versão Final**

Na versão final do sistema, foram adicionadas novas funcionalidades para torná-lo mais completo e prático para o uso do professor. A principal novidade foi a inclusão de um menu com opções, permitindo ao professor escolher o que deseja fazer dentro do sistema, como cadastrar um novo aluno, consultar informações, alterar dados, excluir registros ou visualizar os resultados. Com isso, o sistema passou a ter um CRUD completo — ou seja, permite Cadastrar, Consultar, Atualizar e Deletar os dados dos alunos.

## **Dia 4 – Término do sistema (25/05/2025)**

Após todo o projeto, terminamos de codificar, além disso, terminamos as documentações necessárias para envio ao cliente.

Teste do código:

Menu

```
CSYSTEM: MENU
Quantidade de alunos já cadastrados: 0
1. Cadastrar aluno
2. Atualizar nota
3. Listar alunos
4. Excluir aluno
0. Sair
Escolha uma opção:
```

Fonte: Autoria própria (2025)

Cadastrar

```
CSYSTEM - Cadastrar
Digite o nome do aluno: Robson
Digite o RGM do aluno: 12345678
Digite a nota da Avaliação 1 (0 a 5): 3
Digite a nota da Avaliação 2 (0 a 5): 2
media: 5,0 - Fazer prova AF
Digite a nota da Avaliação Final (0 A 5): 3
Nova media do aluno: 6,0 - Aprovado!
Pressione qualquer tecla para continuar. . . |
```

Fonte: Autoria própria (2025)

Atualizar

```
Digite o RGM do aluno que deseja atualizar:
12345678
Atualizando notas do aluno Robson (RGM 12345678)
Digite a nova nota da Avaliação 1 (0 a 5): 3
Digite a nova nota da Avaliação 2 (0 a 5): 3
Media: 6,0 - Aprovado!
```

Fonte: Autoria própria (2025)

## Listar

```
CSYSTEM - Listagem
Lista de alunos

Aluno 1
Nome: Robson
RGM: 12345678
Nota Final: 6,0
Situação: Aprovado
Pressione qualquer tecla para continuar. . . |
```

Fonte: Autoria própria (2025)

## Excluir

```
CSYSTEM - Excluir
Digite o RGM do aluno que deseja excluir:
12345678
Aluno com o RGM 12345678 excluído com sucesso.
```

Fonte: Autoria própria (2025)

## Sair

```
-----
Process exited after 1103 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Fonte: Autoria própria (2025)

## DIA 5 – Aperfeiçoamento do código (26/05/2025)

O grupo se reuniu pela última vez para melhorar o código e treinar para a apresentação do dia 27/05 diante da gestora/cliente.



## APROVA+

### Português Estruturado

1. Digitar “1” para cadastrar Aluno;

1.1 Escrever Nome;

1.2 Escrever RGM;

1.3 Inserir ‘nota’ <A1>;

1.4 Inserir ‘nota’ <A2>;

1.5 Somar notas;

1.6 Se a somatória for <maior> que **5**, aluno ‘aprovado’;

1.7 Se a somatória for <menor> que **5**, aluno ‘reprovado’;

1.8 Caso o <aluno> for ‘reprovado’, precisará fazer <AF>;

1.9 Somar ‘nota’ da <A1> com <AF>;

1.10 Somar ‘nota’ da <A2> com <AF>;

1.11 Se a somatória da <AF> for maior que **6**, aluno aprovado;

1.12 Se a somatória da <AF> for menor que **6**, aluno reprovado;

2. Digitar “2” para Atualizar Nota;

2.1 Colocar RGM cadastrado;

2.3 Inserir ‘nota’ <A1>;

2.4 Inserir ‘nota’ <A2>;

2.5 Somar notas;

2.6 Se a somatória for <maior> que **5**, aluno ‘aprovado’;

2.7 Se a somatória for <menor> que **5**, aluno ‘reprovado’;

2.8 Caso o <aluno> for ‘reprovado’, precisará fazer <AF>;

2.9 Somar ‘nota’ da <A1> com <AF>;

2.10 Somar ‘nota’ da <A2> com <AF>;

2.11 Se a somatória da <AF> for maior que **6**, aluno aprovado;

2.12 Se a somatória da <AF> for menor que **6**, aluno reprovado;

3. Digitar “3” para Listar todos os Alunos cadastrados;
4. Digitar “4” para Excluir Alunos;
  - 4.1 Digitar RGM que deseja excluir;
5. Digitar “0” para encerrar o programa;

## Linguagem C

```
#include <stdio.h>
#include <string.h>
#include <locale.h>
#include <stdlib.h>

// Dados do aluno
typedef struct //Typedef serve para não escrever struct para declarar a variável
{
    char Nome_Alu [50];
    int RGM;
    float A1, A2, AF, Media_Alu;
} Aluno;

// Criei a variável alunos
Aluno alunos[100];

// Dados da sala
typedef struct
{
    float Media_sala;
```



```
        int Total_Alu;
    } Sala;

    // Criei a variável sala
    Sala sala;

    // Opção para caso o aluno seja aprovado ou reprovado
    typedef struct
    {
        int valor;
        int menu;
        int executar;
    } Valores;

    // Criei a variável opcao
    Valores valores;

    int main(){

        setlocale(LC_ALL, "Portuguese");

        sala.Total_Alu =0;
        valores.executar = 1;

        // Funções CRUD
        while (valores.executar) {
            printf("\nCSYSTEM: MENU\n");
            printf("Quantidade de alunos já cadastrados: %d\n", sala.Total_Alu);
            printf("1. Cadastrar aluno\n");
            printf("2. Atualizar nota\n");
```

```
printf("3. Listar alunos\n");  
printf("4. Excluir aluno\n");  
printf("0. Sair\n");  
printf("Escolha uma opção: ");  
scanf("%d", &valores.menu);  
printf("\n");  
system("pause");  
system("cls");
```

```
switch(valores.menu)
```

```
{
```

```
    case 0:
```

```
    {
```

```
        valores.executar = 0;
```

```
    }
```

```
    break;
```

```
    case 1:
```

```
    {
```

```
        printf("CSYSTEM - Cadastrar \n");
```

```
        // Caso tenha igual ou mais alunos
```

cadastrados

```
        if(sala.Total_Alu >= 100)
```

```
        {
```

```
            printf("Limite de alunos atingidos\n");
```

```
            // Caso para cadastrar o aluno
```

```
        } else
```

```
        {
```

```
            // Pedir as informações do aluno
```

(Nome, RGM, A1, A2 e caso precise de AF)

```
printf("Digite o nome do aluno: ");  
getchar();  
  
fgets(alunos[sala.Total_Alu].Nome_Alu,  
sizeof(alunos[sala.Total_Alu].Nome_Alu), stdin);  
  
alunos[sala.Total_Alu].Nome_Alu[strcspn(alunos[sala.Total_Alu].Nome_A  
lu, "\n")] = 0; //Para pular linha após o fgets  
  
//alunos[sala.Total_Alu].Nome_Alu: onde o nome será armazenado.  
//sizeof: garante que não será  
lido mais do que o tamanho permitido no vetor.  
//stdin: lê da entrada padrão  
(teclado).  
  
int RGM_novo;  
int RGM_duplicado;  
  
do {  
    RGM_duplicado = 0;  
    printf("Digite o RGM do aluno:  
");  
    scanf("%d", &RGM_novo);  
  
    for (int RGM_pos = 0;  
        RGM_pos < sala.Total_Alu; RGM_pos++)  
        if (alunos[RGM_pos].RGM  
            == RGM_novo)  
            {  
                printf("Erro: Já existe um  
aluno com esse RGM.\n");
```

```

RGM_duplicado = 1;
break;
    }
}
} while (RGM_duplicado);

alunos[sala.Total_Alu].RGM =
RGM_novo;

do{
    printf("Digite a nota da Avaliação 1 (0 a
5): ");

    scanf("%f", &alunos[sala.Total_Alu].A1);
    } while (alunos[sala.Total_Alu].A1
< 0 || alunos[sala.Total_Alu].A1 > 5);

do{
    printf("Digite a nota da Avaliação 2 (0 a
5): ");

    scanf("%f", &alunos[sala.Total_Alu].A2);
    } while (alunos[sala.Total_Alu].A2
< 0 || alunos[sala.Total_Alu].A2 > 5);

    // Média do aluno
    alunos[sala.Total_Alu].Media_Alu
= alunos[sala.Total_Alu].A1 + alunos[sala.Total_Alu].A2;

    // Opções caso o aluno tenha a
média maior ou menor que 6

    if(alunos[sala.Total_Alu].Media_Alu >=6)
    {
```

```
printf("Media: %.1f - Aprovado!\n",
alunos[sala.Total_Alu].Media_Alu);

    } else
    {
        printf("media:
        %.1f - Fazer prova AF\n", alunos[sala.Total_Alu].Media_Alu);
        do {
            printf("Digite a nota
            da Avaliação Final (0 A 5): ");
            scanf("%f",
            &alunos[sala.Total_Alu].AF);
        }while
        (alunos[sala.Total_Alu].AF < 0 || alunos[sala.Total_Alu].AF > 5);

        // Substituir a menor nota
        pela AF
        if
        (alunos[sala.Total_Alu].A1 < alunos[sala.Total_Alu].A2)
        {
            alunos[sala.Total_Alu].A1 = alunos[sala.Total_Alu].AF;
        } else
        {
            alunos[sala.Total_Alu].A2 = alunos[sala.Total_Alu].AF;
        }

        // Nova média do aluno

        alunos[sala.Total_Alu].Media_Alu = alunos[sala.Total_Alu].A1 +
        alunos[sala.Total_Alu].A2;

        // Opções caso
        novamente seja reprovado ou aprovado
```



```
    }

    int UpRGM, encontrado = 0;
    printf("Digite o RGM do aluno que deseja
atualizar: \n");

    scanf("%d", &UpRGM);

    for(int UpAlu = 0; UpAlu < sala.Total_Alu;
UpAlu++)
    {
        if(alunos[UpAlu].RGM == UpRGM)
        {
            encontrado = 1;
            printf("Atualizando notas do
aluno %s (RGM %d)\n", alunos[UpAlu].Nome_Alu, alunos[UpAlu].RGM);
            sala.Media_sala -=
alunos[UpAlu].Media_Alu;

            do {
                printf("Digite a nova nota da Avaliação 1 (0 a 5):
");
                scanf("%f", &alunos[UpAlu].A1);
            } while (alunos[UpAlu].A1 < 0 || alunos[UpAlu].A1 >
5);

            do {
                printf("Digite a nova nota da Avaliação 2 (0 a 5):
");
                scanf("%f", &alunos[UpAlu].A2);
            } while (alunos[UpAlu].A2 < 0 || alunos[UpAlu].A2 >
5);
```

```
alunos[UpAlu].Media_Alu = alunos[UpAlu].A1 +  
alunos[UpAlu].A2;  
  
if(alunos[UpAlu].Media_Alu >=6)  
{  
    printf("Media: %.1f -  
Aprovado!\n", alunos[UpAlu].Media_Alu);  
}  
else  
{  
    printf("media: %.1f -  
Fazer prova AF\n", alunos[UpAlu].Media_Alu);  
do {  
    printf("Digite  
a nota da Avaliação Final (0 A 5): ");  
    scanf("%f",  
&alunos[UpAlu].AF);  
}while  
(alunos[UpAlu].AF < 0 || alunos[UpAlu].AF > 5);  
  
// Substituir a  
menor nota pela AF  
if (alunos[UpAlu].A1  
< alunos[UpAlu].A2)  
{  
    alunos[UpAlu].A1 = alunos[UpAlu].AF;  
}  
else  
{  
    alunos[UpAlu].A2 = alunos[UpAlu].AF;  
}
```





```
break;
```

```
//Listar alunos
```

```
case 3:
```

```
{
```

```
    printf("CSYSTEM - Listagem\n");
```

```
    if(sala.Total_Alu == 0)
```

```
    {
```

```
        printf("Nenhum aluno cadastrado\n");
```

```
    } else
```

```
    {
```

```
        printf("Lista de alunos \n");
```

```
        valores.valor = 0;
```

```
        for (valores.valor; valores.valor <
```

```
sala.Total_Alu; valores.valor++)
```

```
        {
```

```
            printf("\nAluno %d \n",
```

```
valores.valor+1);
```

```
            printf("Nome: %s \n",
```

```
alunos[valores.valor].Nome_Alu);
```

```
            printf("RGM: %d \n",
```

```
alunos[valores.valor].RGM);
```

```
            printf("Nota Final: %.1f \n",
```

```
alunos[valores.valor].Media_Alu);
```

```
            if
```

```
(alunos[valores.valor].Media_Alu >= 6)
```

```
            {
```

```
                printf("Situação:
```

```
Aprovado \n");
```

```
            } else
```

```
            {
```

```
printf("Situação: Reprovado \n");  
  
        }  
    }  
}  
system("pause");  
system("cls");  
}  
break;  
  
//Excluir aluno  
case 4:  
    {  
        int ExcluRGM, encontrado = 0;  
        printf("\nCSYSTEM - Excluir\n");  
        printf("Digite o RGM do aluno que deseja  
excluir: \n");  
  
        scanf("%d", &ExcluRGM);  
  
        for(int atual = 0; atual < sala.Total_Alu;  
atual++)  
        {  
            if(alunos[atual].RGM == ExcluRGM)  
            {  
                encontrado = 1;  
  
                sala.Media_sala -=  
alunos[atual].Media_Alu;  
  
                for (int novo = atual; novo <  
sala.Total_Alu - 1; novo++)  
                {
```

```
alunos[novo] = alunos  
[novo + 1];  
}  
  
sala.Total_Alu--;  
printf("Aluno com o RGM %d excluído com  
sucesso.\n", ExcluRGM);  
break;  
}  
}  
  
if(!encontrado)  
{  
    printf("Aluno com RGM %d não  
encontrado.\n", ExcluRGM);  
}  
system("pause");  
system("cls");  
}  
break;  
}  
}  
return 0;  
}
```



# CSystem

## O futuro com inovação

# Quem somos?

A CSystem fornece sistemas em linguagem de programação C, é fundamental implementar a tecnologia para auxiliar nas tarefas diárias. Para inovar e automatizar processos a CSystem cria os sistemas de forma eficiente e com qualidade.



## APROVA+

## Pedido APROVA+

Calcular a média dos alunos da Instituição Braz Cubas. Utilizando de as provas A1 e A2 para calcular a média.

E se a média for  $\geq 6$  aluno aprovado

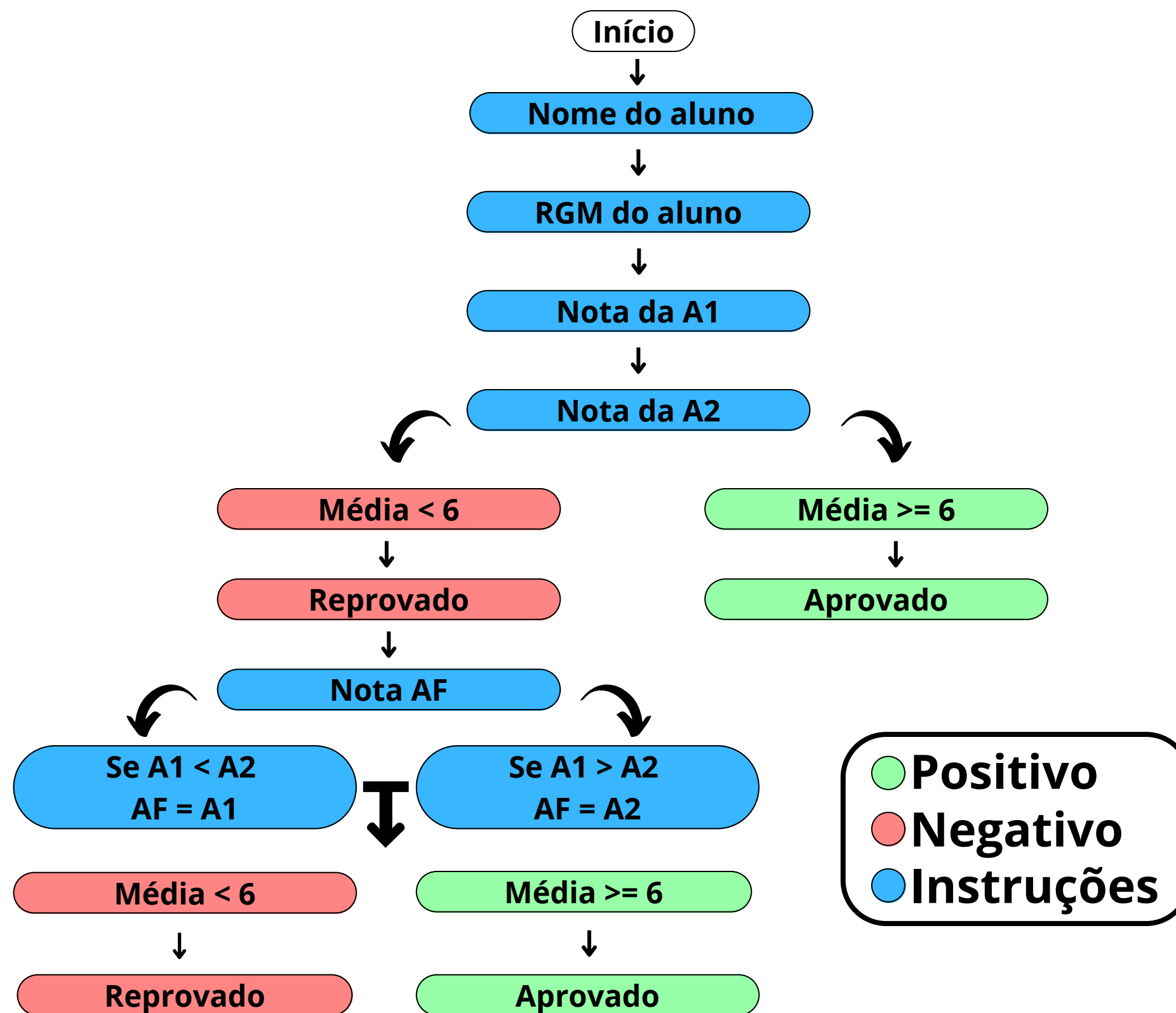
Se a média for menor que 6 o aluno precisa fazer a AF que substituirá a menor nota entre A1 e A2.

Deve-se ter um menu de navegação

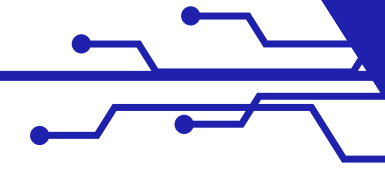
## Benefícios

O sistema facilita o controle de notas, permitindo à professora Renata Costa gerenciar informações com rapidez e precisão pelo RGM. Com interface simples e cálculo automático de médias, reduz erros, evita duplicidades e moderniza os processos acadêmicos da instituição.

# Fluxograma de sistema de cadastro







## Linguagem utilizada:

- Desenvolvido em linguagem C.
- Auxilia na compreensão da lógica dos algoritmos e do funcionamento interno de um programa.

## Organização dos dados:

- Uso de structs

## Controle de fluxo do sistema:

- Utilização de estruturas de decisão:
  - if
  - switch/case
- Uso de laços de repetição:
  - for
  - while
  - do/while

## Funções de entrada e saída:

- **scanf** para entrada de dados.
- **printf** para exibição de informações.

## Bibliotecas utilizadas:

- **stdio.h** – Operações de entrada e saída padrão.
- **string.h** – Manipulação de strings.
- **locale.h** – Configuração de idioma para permitir acentuação correta e caracteres especiais.
- **stdlib.h** – Utilizada para funções utilitárias como: pause e limpar terminal

## Funcionalidades do sistema (CRUD):

- **Cadastro de alunos.**
- **Realização de consultas.**
- **Update (atualização das notas).**
- **Deletar registros de alunos.**

```
CSYSTEM: MENU
Quantidade de alunos já cadastrados: 0
1. Cadastrar aluno
2. Atualizar nota
3. Listar alunos
4. Excluir aluno
0. Sair
Escolha uma opção:
```

Será solicitada uma das opções para prosseguir

- Cadastrar aluno
- Atualizar nota
- Listar alunos
- Excluir aluno
- Sair

```
CSYSTEM - Cadastrar
Digite o nome do aluno: Robson
Digite o RGM do aluno: 12345678
Digite a nota da Avaliação 1 (0 a 5): 3
Digite a nota da Avaliação 2 (0 a 5): 2
media: 5,0 - Fazer prova AF
Digite a nota da Avaliação Final (0 A 5): 3
Nova media do aluno: 6,0 - Aprovado!
Pressione qualquer tecla para continuar. . . |
```

Após escolher a opção 1, serão solicitadas as seguintes informações:

- Nome
- RGM
- A1
- A2
- AF (Caso a media seja menor que 6)

```
Digite o RGM do aluno que deseja atualizar:
12345678
Atualizando notas do aluno Robson (RGM 12345678)
Digite a nova nota da Avaliação 1 (0 a 5): 3
Digite a nova nota da Avaliação 2 (0 a 5): 3
Media: 6,0 - Aprovado!
```

Após escolher a opção 2, serão solicitadas as seguintes informações:

- RGM
- A1
- A2
- AF (Caso a media seja menor que 6)

# Demonstração do Sistema

```
CSYSTEM - Listagem  
Lista de alunos
```

```
Aluno 1  
Nome: Robson  
RGM: 12345678  
Nota Final: 6,0  
Situação: Aprovado  
Pressione qualquer tecla para continuar. . . |
```

Após escolher a opção 3, serão informadas as seguintes informações:

- Nome
- RGM
- Media Final
- Situação

```
CSYSTEM - Excluir
```

```
Digite o RGM do aluno que deseja excluir:  
12345678  
Aluno com o RGM 12345678 excluído com sucesso.
```

Após escolher a opção 4, será solicitado o RGM do aluno que deseja remover

- RGM

```
-----  
Process exited after 1103 seconds with return value 0  
Pressione qualquer tecla para continuar. . .
```

Após escolher a opção 0, o sistema será encerrado

# Possíveis Melhorias

## **Criação de uma interface:**

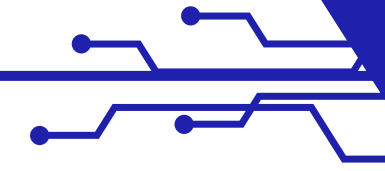
- **Tornado projeto mais intuitivo, organizado e agradável para o usuário.**

## **Compactação do Código:**

- **Melhorando assim cada vez mais o desempenho do sistema e também reduzindo o tempo de carregamento.**

## **Banco de dados:**

- **Permitindo salvar as informações do usuário em arquivos, garantindo que os dados sejam mantidos mesmo após o programa ser fechado.**



**Este projeto em linguagem C permitiu aplicar conceitos de programação estruturada, como structs, vetores e controle de fluxo, na criação de um sistema de cadastro e controle de notas de alunos. Destacou-se a importância da validação de dados e da organização lógica do código. Além disso, reforçou-se a necessidade de um bom planejamento para facilitar futuras melhorias, como a inclusão de armazenamento em arquivos e relatórios detalhados.**



**Obrigado pela atenção**



## Referências

YOUTUBE. *Linguagem C - Começando do absoluto zero*. 2022. Disponível em: [https://youtu.be/2w8GYzBJNj8?si=5knKlrXH6SNTQ\\_vP](https://youtu.be/2w8GYzBJNj8?si=5knKlrXH6SNTQ_vP). Acesso em: 10 abril 2025.

YOUTUBE. *Linguagem C - Aula 4.1.1 - Operações lógicas: disjunção, conjunção e negação*. 2022. Disponível em: <https://youtu.be/uGkX7tdqV8A?si=bGY7PCObfi-B1K24>. Acesso em: 15 abril 2025.

YOUTUBE. *Linguagem C - Aula 4.3 - Conheça o comando SWITCH-CASE - estrutura de decisão múltipla*. 2022. Disponível em: <https://youtu.be/xgwrJNZGo24?si=6oJNBpeykfKznEcW>. Acesso em: 29 abril 2025.

YOUTUBE. *Linguagem C - Aula 5.4 - Conheça os comandos de Controle de Fluxo break e continue*. 2022. Disponível em: <https://youtu.be/iWYLDyX6OsY?si=He1TqVOej7Cu0z96>. Acesso em: 29 abril 2025.

YOUTUBE. *Linguagem C - Aula 6.1 - Domine vetores (arrays) em linguagem C*. 2022. Disponível em: <https://youtu.be/5fSf9xSJK7c?si=SRZKEGsDq94z-F1A>. Acesso em: 29 abril 2025.

YOUTUBE. *Linguagem C - Aula 6.2 - Strings: trabalhando com dados de texto em C*. 2022. Disponível em: <https://youtu.be/4x7Qht5Dmcw?si=A5-xElf-TV8BK0ZS>. Acesso em: 14 maio 2025.

YOUTUBE. *Linguagem C - Aula 8 - Estude Structs (registros) em C*. 2022. Disponível em: <https://youtu.be/-SgbvUL2UVc?si=xYPvQxIBVnSqXVET>. Acesso em: 14 maio 2025.

YOUTUBE. *Linguagem C - Aula 9.1 - Aprenda a criar e usar Funções e Procedimentos em C*. 2022. Disponível em: [https://youtu.be/jd9PABblvbA?si=rBSkOmv\\_ToSYJHhe](https://youtu.be/jd9PABblvbA?si=rBSkOmv_ToSYJHhe). Acesso em: 17 maio 2025.

YOUTUBE. *O Que é CRUD na PROGRAMAÇÃO O guia definitivo do zero*. 2025. Disponível em: <https://youtu.be/dfDCq2sNeHE?si=nF-QepApa7KFN2Y->. Acesso em: 18 maio 2025.

YOUTUBE. *Aplicando CRUD*. Canal Como Programar em C. Disponível em: <https://www.youtube.com/@Comoprogramaremc>. Acesso em: 18 maio 2025.