

GCENTRO UNIVERSITÁRIO
CIÊNCIA DA COMPUTAÇÃO / ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS
PROGRAMAÇÃO DE COMPUTADORES

Projeto Sistema de Registro de notas

GRUPO 3

Eduardo Felipe Braga Silva
Hugo Trindade Mendes Cortiz
Henrique Gobetti Guedes
Lucas Henrique de Moraes Silva
Anderson Marquetti

Mogi das Cruzes, Maio de 2025

RESUMO

Este projeto visa criar um programa tendo base a linguagem C para registro, armazenamento e exibição das notas semestrais de até 100 alunos em uma instituição de ensino fictícia

Alunos:

Eduardo Felipe Braga Silva.
Lucas Henrique de Moraes Silva
Henrique Gobetti Guedes
Hugo Trindade Mendes Cortiz

Orientador:

Renata, Prof.
Centro Universitário

Palavras-chave: Abnt, Programação de Computadores, Dados, cálculo

Área do conhecimento: Programação de Computadores

ABSTRACT

This Project aims to create an application based in the C programming language for registry, storage and exhibition of Half Early grades of up to a hundred students on an institution

Keywords: Computer programming, *Data, calculation*

SUMÁRIO

Sumário

ABSTRACT	2
SUMÁRIO.....	3
INTRODUÇÃO	4
GITHUB.....	4
PROPOSTA E EXECUÇÃO	4
EXPLICANDO O CÓDIGO	5
EMPRESA SCHOOL LOGS COMPANY	12
CONCLUSÃO.....	13
REFERÊNCIAS.....	15

INTRODUÇÃO

Ao buscarmos entender a maneira mais coesa de compreender os processos lógicos que explicam o funcionamento de códigos e execução de instruções pelas máquinas tal qual o computador, é essencial que o futuro profissional possua um conhecimento completo e robusto a respeito dos pilares do Pensamento Computacional, estes sendo, Abstração, Decomposição, Algoritmo e Reconhecimento de padrões, neste trabalho decidimos utilizar o clássico jogo de tabuleiro Damas como base para este processo de aprendizado .

GITHUB

Todo o projeto foi atualizado em 2 bancos de dados, o mais recente estará conectado via Hyperlink neste documento, nele, há o código mais recente desenvolvido pela equipe.

[Link Github](#)

PROPOSTA E EXECUÇÃO

Foi pedido pelo cliente, inicialmente, que fosse elaborado um programa que pudesse registrar notas de alunos e realizar o cálculo somatório das notas de modo a checar se a soma das notas superaria a nota MÉDIA de valor 5, caso superasse o ALUNO seria atribuído a condição APROVADO, caso contrário a condição seria REPROVADO.

Conforme se passaram as semanas outros quesitos foram requisitados pelo cliente, como a implementação de variáveis do tipo VETOR para armazenamento de mais alunos, 100 para ser exato, tal funcionalidade foi implementada.

Com isso surgiu a necessidade de poder consultar os alunos que fossem registrados neste “Banco de dados local”, então, elaboramos uma função para consultar uma tabela com os dados dos alunos.

Com isso precisávamos de uma forma de navegar entre essas opções, então surgiu a ideia e a necessidade da criação de um menu de usuário no qual o ADMIN poderia navegar, implementamos.

Por fim, como medida experimental, implementamos uma seção de LOGIN, onde o usuário pode entrar em sua conta com um usuário e senha, e consultar seus alunos

com privacidade, sem contar também com os elementos visuais de MENU feitos com ASCII art.

EXPLICANDO O CÓDIGO

```
1  #include <iostream>
2  #include <iomanip>
3  #include <string>
4  #include <vector>
5  #include <chrono>
6  #include <thread>
```

Figura: Livrarias usadas no código

Em um código C os primeiros parâmetros a serem estabelecidos são as livrarias da linguagem que adicionam comandos e funcionalidades extras ao ambiente de programação, as livrarias utilizadas foram:

Então temos o uso de “namespace std” para estabelecer a estrutura dos prints e register usadas no código.

```
8  using namespace std;
9
10 struct Aluno {
11     string nome;
12     string rgm;
13     float A1 = 0, A2 = 0, AF = 0;
14 };
15
```

Figura: Namespace e Struct

Em seguidas temos a criação a estrutura “struct Aluno”, estabelecendo uma variável matriz de nome “Aluno” que encapsula 3 variáveis, “string nome”, “string rgm”, “float A1”, “float A2”, “float AF”. Cada variável com a estrutura “A + numeral ou F” correspondem a uma atividade ou prova executada pelo aluno que corresponde a uma nota.

```

void Login() {
    cout << "          " << endl;
    cout << "      ( ) " << endl;
    cout << "      " << endl;
    cout << "Login" << endl;
    cout << "      " << endl;
    cout << "      " << endl;
    cout << "      " << endl;
    string usuario, senha;
    const string usuarioCorreto = "Admin";
    const string senhaCorreta = "1234";

    while (true) {
        cout << "====REALIZE O LOGIN ====\\n";
        cout << "Usuario: ";
        cin >> usuario;
        cout << "Senha: ";
        cin >> senha;

        if (usuario == usuarioCorreto && senha == senhaCorreta) {
            cout << "Login realizado com sucesso!\\n";
            system("pause");
            system("cls");
            break;
        } else {
            cout << "Usuario ou senha incorretos. Tente novamente.\\n\\n";
        }
    }
}

```

Figura: Comando de login

Aqui temos a estrutura de um comando separado do MAIN chamado "Login" nele desenhamos uma logo exibimos uma estrutura visual de usuário e senha e pedimos um input de valores a serem atribuídos ao usuário e a senha respectivamente.


```

110 int main() {
111     Login();
112
113     int opcao;
114     vector<Aluno> alunos;
115     Aluno aluno;
116
117     Logo();
118     std::this_thread::sleep_for(std::chrono::milliseconds(3000));
119     system("cls");
120
121     do {
122         system("cls");
123         cout << "                                "<< endl;
124         cout << "                                "<< endl;
125         cout << "                                "<< endl;
126         cout << "                                "<< endl;
127         cout << "                                "<< endl;
128         cout << "                                "<< endl;
129         cout << "\n Menu \n";
130         cout << "1. Tabela de Notas\n";
131         cout << "2. Incluir Notas\n";
132         cout << "3. Filtrar\n";
133         cout << "4. Creditos\n";
134         cout << "5. Sair\n";
135         cout << "Escolha uma opcao: ";
136         cin >> opcao;

```

Figura: Começo de método MAIN

Aqui temos o início do corpo do programa, o main, neste trecho, criamos variáveis que serão integrais para o funcionamento do sistema, evocamos o método logo, e desenhamos o MENU, que será interagido por meio do valor inserido dentro de um switch case.


```

50 void EditarAluno(vector<Aluno>& alunos) {
51     string rgm;
52     cout << "\nInsira o RGM do Aluno que deseja editar: ";
53     cin >> rgm;
54
55     for (auto& aluno : alunos) {
56         if (aluno.rgm == rgm) {
57             cin.ignore();
58
59             cout << "\nDigite o novo nome (ou pressione ENTER para nao alterar): ";
60             string novoNome;
61             getline(cin, novoNome);
62             if (!novoNome.empty()) {
63                 aluno.nome = novoNome;
64             }
65
66             cout << "\nDigite o novo RGM do aluno: ";
67             string novoRGM;
68             getline(cin, novoRGM);
69             while (novoRGM.length() != 8) {
70                 cout << "\nInsira um RGM valido de 8 digitos: ";
71                 getline(cin, novoRGM);
72             }
73             aluno.rgm = novoRGM;
74
75             cout << "\nInsira a nova nota A1: ";
76             float novaNotaA1;
77             cin >> novaNotaA1;
78             while (novaNotaA1 < 0 || novaNotaA1 > 5) {
79                 cout << "Insira uma nota valida (0-5): ";
80                 cin >> novaNotaA1;
81             }
82             aluno.A1 = novaNotaA1;
83
84             cout << "\nInsira a nova nota A2: ";
85             float novaNotaA2;
86             cin >> novaNotaA2;
87             while (novaNotaA2 < 0 || novaNotaA2 > 5) {
88                 cout << "Insira uma nota valida (0-5): ";
89                 cin >> novaNotaA2;
90             }
91             aluno.A2 = novaNotaA2;
92
93             cout << "\nInsira a nova nota AF: ";
94             cin >> aluno.AF;

```

Figura: Função de editar dados de um aluno

```

95
96     float novaMedia = (aluno.A1 < aluno.A2 ? aluno.AF + aluno.A2 : aluno.A1 + aluno.AF);
97     string situacao = novaMedia >= 6 ? "Aprovado c/ AF" : "Reprovado";
98
99     system("cls");
100     cout << "\nSituacao do Aluno: " << situacao << endl;
101     cout << "\nNome: " << aluno.nome
102         << " - Nota A1: " << aluno.A1
103         << " - Nota A2: " << aluno.A2
104         << " - Situacao: " << situacao << endl;
105     return;
106 }
107
108 cout << "\nAluno nao encontrado." << endl;
109 }

```

Figura: Substituição de média, situação e checagem de RGM

Nesta seção do código é feita a função de substituir alunos baseado no número de seu RGM, pedindo um input novo para cada dado.

```

138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
...

switch (opcao) {
case 1:
    if (alunos.empty()) {
        cout << "Nenhum aluno cadastrado.\n";
        system("pause");
    } else {
        cout << left << setw(15) << "Nome"
            << setw(10) << "rgm"
            << setw(10) << "A1"
            << setw(10) << "A2"
            << setw(10) << "AF"
            << setw(15) << "Situacao" << endl;

        for (const Aluno& a : alunos) {
            float media = a.A1 + a.A2;
            string situacao;

            if (media >= 6) {
                situacao = "Aprovado";
            } else {
                float novaMedia = (a.A1 < a.A2 ? a.AF + a.A2 : a.A1 + a.AF);
                situacao = novaMedia >= 6 ? "Aprovado c/ AF" : "Reprovado";
            }

            cout << left << setw(15) << a.nome
                << setw(10) << a.rgm
                << setw(10) << a.A1
                << setw(10) << a.A2
                << setw(10) << a.AF
                << setw(15) << situacao << endl;
        }
    }
}
cout << "\nPressione ENTER para retornar ao menu.";
cin.ignore();
cin.get();
system("cls");
break;

```

Figura: Switch case 1

Observamos nesta figura o caso de número “1” do switch, nesta opção demos print em uma planilha com todas as informações que foram em algum momento inseridas no sistema.

```

175
176
177 case 2:
178     cout << "Digite o nome do aluno:";
179     cin.ignore(1000, '\n');
180     getline(cin, aluno.nome);
181
182     cout << "Digite o RGM do aluno (maximo 8 caracteres): ";
183     getline(cin, aluno.rgm);
184
185     while (aluno.rgm.length() > 8 || aluno.rgm.length() < 8 || aluno.rgm.empty()) {
186         cout << "RGM invalido. Insira um RGM entre 1 e 8 caracteres:";
187         getline(cin, aluno.rgm);
188     }
189
190     cout << "Digite a nota A1: ";
191     cin >> aluno.A1;
192     while (aluno.A1 < 0 || aluno.A1 > 5) {
193         cout << "Digite A1 novamente: ";
194         cin >> aluno.A1;
195     }
196
197     cout << "Digite a nota A2: ";
198     cin >> aluno.A2;
199     while (aluno.A2 < 0 || aluno.A2 > 5) {
200         cout << "Digite A2 novamente: ";
201         cin >> aluno.A2;
202     }
203
204     if ((aluno.A1 + aluno.A2) < 6) {
205         cout << "Voce foi reprovado. Digite a nota da AF: ";
206         cin >> aluno.AF;
207     } else {
208         aluno.AF = 0;
209     }
210
211     alunos.push_back(aluno);
212     system("cls");
213     break;
214

```

Figura: Switch case 2

No case 2 realizamos a funcionalidade principal do programa, que é registrar as notas dos alunos e determinar se eles foram aprovados ou não.

```

216
217 case 3:
218     system("cls");
219     Filtrar(alunos);
220     break;

```

Figura: Case 3

Executamos o comando externo filtrar.

```

229
230 case 5:
231     cout << "Saindo...";
232     break;

```

Figura: Case 4

Fechamos o programa.

```

234 |         default:
235 |             cout << "Opcao invalida.\n";
236 |
237 |         }
238 |
239 |     } while (opcao != 5);
240 |
241 |     return 0;
242 | }

```

Figura: Default e do método main

Aqui ditamos se a opção inserida é válida. Então fechamos o método main.

EMPRESA SCHOOL LOGS COMPANY

A

empresa



School Logs Company é uma empresa com o foco na elaboração de sistemas para instituições de educação de diferentes níveis de educação.

CONCLUSÃO

Concluíse que tudo que foi requisitado pelo cliente em reuniões foi cumprido além de melhorias não premeditadas pela instituição, tais melhorais transformam a UX (User Experience) do programa mais trabalhada melhorando de modo extremo a facilidade de uso.

```
aluno.rgn = nomeo);
cout << "Insira a nova nota A1: ";
float novanota1;
cin >> novanota1;
while (novanota1 < 0 || novanota1 > 5) {
    cout << "Insira uma nota valida (0-5): ";
    cin >> novanota1;
}
aluno.A1 = novanota1;

cout << "Insira a nova nota A2: ";
float novanota2;
cin >> novanota2;
while (novanota2 < 0 || novanota2 > 5) {
    cout << "Insira uma nota valida (0-5): ";
    cin >> novanota2;
}
aluno.A2 = novanota2;

cout << "Insira a nova nota AF: ";
cin >> aluno.AF;

float novamedia = (aluno.A1 + aluno.A2 + aluno.AF + aluno.A2 + aluno.A1 + aluno.AF);
string situacao = novamedia >= 6 ? "Aprovado c/ AF" : "Reprovado";

system("cls");
cout << "Situacao do aluno: " << situacao << endl;
cout << "Nome: " << aluno.nome
<< " - Nota A1: " << aluno.A1
<< " - Nota A2: " << aluno.A2
<< " - Situacao: " << situacao << endl;
return;
}

cout << "Aluno nao encontrado." << endl;
}

int Filtrar(const vector<Aluno> &alunos) {
    int opcaoFiltro;
    cout << "Ordernar por...: ";
    cout << "1-1. Aprovados";
    cout << "2-2. Reprovados";
    cout << "3-3. Woflar";
    cout << "Insira uma opcao: ";
    cin >> opcaoFiltro;

    system("cls");

    if (alunos.empty()) {
        cout << "Nenhum aluno cadastrado.\n";
        system("pause");
        return opcaoFiltro;
    } // combinacao da funcao reservada da string que verifica se o input está vazio com a checagem da memória temporaria, verifica se caso esteja vazio a tabela de alunos, printa que nao tem aluno, pausa o sistema até receber um input de continuacao e retorna a função opcaoFiltr

    cout << left << setw(15) << "Nome"
    << setw(15) << "A1"
    << setw(15) << "A2"
    << setw(15) << "AF"
    << setw(15) << "Situacao" << endl;
    // recebe informações input do usuário em ordem, e nessa ordem classifica cada input correspondente as variáveis, sendo nome, a1, a2, af, situacao.

    for (const Aluno& a : alunos) {
        float media = a.A1 + a.A2;
    }
}
```

```
cin >> opcao;
// cria um loop de repetição que tem como principio o print do menu, tendo 6 opções, dependendo do input (6) o console retorna uma função específica, cada uma dessas funções está armazenado em um switch case, o input recebido será declarado pelo computador como a opção e

switch (opcao) {
    case 1:
        if (alunos.empty()) {
            cout << "Nenhum aluno cadastrado.\n";
            system("pause");
        } else {
            cout << left << setw(15) << "Nome"
            << setw(15) << "A1"
            << setw(15) << "A2"
            << setw(15) << "AF"
            << setw(15) << "Situacao" << endl;

            for (const Aluno& a : alunos) {
                float media = a.A1 + a.A2;
                string situacao;

                if (media >= 6) {
                    situacao = "Aprovado";
                } else {
                    float novamedia = (a.A1 + a.A2 + a.AF + a.A2 + a.A1 + a.AF);
                    situacao = novamedia >= 6 ? "Aprovado c/ AF" : "Reprovado";
                }

                cout << left << setw(15) << a.nome
                << setw(15) << a.rgn
                << setw(15) << a.A1
                << setw(15) << a.A2
                << setw(15) << a.AF
                << setw(15) << situacao << endl;
            }

            cout << "\nPressione ENTER para retornar ao menu.";
            cin.ignore();
            cin.get();
            break;
        }

    // o código começa verificando se tem alunos cadastrados na tabela, caso não tenha, então ele vai printar dizendo que ninguém foi cadastrado, pausa o sistema e então retorna ao menu. caso tenha alunos cadastrados, retorna uma tabela contendo o nome dele, rgn, nota e situacao, e

    case 2:
        cin.ignore();
        cout << "Digite o nome do aluno: ";
        getline(cin, aluno.nome);

        cout << "Digite o rgn do aluno (8 caracteres): ";
        getline(cin, aluno.rgn);
        while (aluno.rgn.length() != 8) {
            cout << "Tem inválido. Insira um RGN com 8 caracteres: ";
            getline(cin, aluno.rgn);
        }

        cout << "Digite a nota A1: ";
        cin >> aluno.A1;
        while (aluno.A1 < 0 || aluno.A1 > 5) {
            cout << "Digite A1 novamente (0-5): ";
            cin >> aluno.A1;
        }

        cout << "Digite a nota A2: ";
        cin >> aluno.A2;
        while (aluno.A2 < 0 || aluno.A2 > 5) {
            cout << "Digite A2 novamente (0-5): ";
            cin >> aluno.A2;
        }
}
```


REFERÊNCIAS

- **Normas ABNT para apresentação de trabalhos científicos.** Wikimedia. Disponível em:
<http://pt.wikipedia.org/wiki/Normas_ABNT_para_apresentação_de_trabalhos_científicos>. Acesso em 04 de abr. de 2012.
- **Regras e Normas da ABNT 2012 para formatação de trabalhos acadêmicos.** Trabalhos ABNT. Disponível em:
<<http://www.trabalhosabnt.com/regras-normas-abnt-formatacao>>. Acesso em 04 de abr. de 2012.
- **Trabalhos acadêmicos: Normas da ABNT.** Firb. Disponível em:
<<http://www.firb.br/abntmonograf.htm>>. Acesso em 04 de abr. de 2012.
- **C Tutorial.** Disponível em: <<https://www.w3schools.com/c/index.php>>.