

Aluno: Léo Willian Kölln
Matricula: 0513227-4

Código Fonte

```
// -----  
// - Por: Léo Willian Kölln  
// - Email: leokolln@gmail.com  
// -----  
  
#include <cstdlib>  
#include <iostream>  
#include <iomanip>  
#include <ctime>  
  
#define ALTURA 20  
#define LARGURA (ALTURA + 1)  
  
using namespace std;  
  
void populaSistema();  
void pivotaParcial(const int);  
int encontraMaiorNaColuna(const int, const int);  
void trocaLinhas(const int, const int);  
void escalonaSistema(const int);  
void defineVariaveis();  
void exhibeResultados();  
void exhibeSistema();  
  
typedef long double valSistema;  
  
valSistema sistema[ALTURA][LARGURA]; // Sistema tipo Expandido  
valSistema erros[ALTURA][LARGURA]; // Sistema tipo Expandido (verificação  
de erros)  
valSistema variaveis[ALTURA]; // Resultados do Sistema  
  
int main(int argc, char *argv[])  
{  
    srand(time(NULL));  
  
    int iTempo = time(NULL);  
  
    populaSistema();  
    //exibeSistema();  
    for(int k = 0; k < ALTURA; k++)  
    {  
        pivotaParcial(k);  
        escalonaSistema(k);  
    }  
    defineVariaveis();  
  
    cout << "==" << "Resultados" << "==" << endl;  
    exhibeResultados();  
  
    cout << setprecision (5);  
    cout << "Tempo de Execucao: " << (time(NULL) - iTempo) << "
```

```

Segundos." << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}

void populaSistema()
{
    for(int i = 0; i < ALTURA; i++)
    {
        for(int j = 0; j < LARGURA; j++)
        {
            sistema[i][j] = (rand()%98 + 1);
            erros[i][j] = sistema[i][j];
        }
    }
}

void pivotaParcial(const int k)
{
    trocaLinhas(k, encontraMaiorNaColuna(k, k));
}

int encontraMaiorNaColuna(const int j, const int k)
{
    int iMaior = k;
    valSistema vMaior = sistema[k][j];
    for(int i = (k + 1); i < ALTURA; i++)
    {
        if(vMaior < sistema[i][j])
        {
            iMaior = i;
            vMaior = sistema[i][j];
        }
    }
    return iMaior;
}

void trocaLinhas(const int iOrigem, const int iDestino)
{
    valSistema itemCop;
    for(int j = 0; j < LARGURA; j++)
    {
        itemCop = sistema[iDestino][j];
        sistema[iDestino][j] = sistema[iOrigem][j];
        sistema[iOrigem][j] = itemCop;
    }
}

void escalonaSistema(const int k)
{
    for(int i = (k + 1); i < ALTURA; i++)
    {
        valSistema buffer = sistema[i][k] / sistema[k][k];
        for(int j = (k + 1); j < LARGURA; j++)
        {
            sistema[i][j] -= (buffer * sistema[k][j]);
        }
    }
}

```

```

        }
        sistema[i][k] = 0.0;
    }
}

void defineVariaveis()
{
    for(int k = ALTURA - 1; k >= 0; k--)
    {
        valSistema buffer = 0;
        for(int j = (k + 1); j < (LARGURA - 1); j++)
        {
            buffer += sistema[k][j] * variaveis[j];
        }
        variaveis[k] = ((sistema[k][(LARGURA - 1)] - buffer) /
sistema[k][k]);
    }
}

void exhibeResultados()
{
    for(int k = 0; k < ALTURA; k++)
    {
        cout << setw(2) << (k + 1) << ": " << setw(13) <<
variaveis[k];

        valSistema buffer = 0;
        for(int j = 0; j < (LARGURA - 1); j++)
        {
            buffer += erros[k][j] * variaveis[j];
        }
        cout << setw(0);
        cout << " - Erro no sistema " << setw(2) << (k + 1) << ": "
<< setw(13) << (erros[k][LARGURA - 1] - buffer) << endl;
    }
}

void exhibeSistema()
{
    for(int i = 0; i < ALTURA; i++)
    {
        for(int j = 0; j < LARGURA; j++)
        {
            if(j == (LARGURA - 1))
            {
                cout << "| ";
            }
            cout << setw(2);
            cout << sistema[i][j] << " ";
        }
        cout << endl;
    }
}

```

Resultados

7 Variáveis

1:	-1.01141	- Erro no sistema	1:	3.1225e-017
2:	0.589295	- Erro no sistema	2:	1.38778e-017
3:	1.34652	- Erro no sistema	3:	1.9082e-017
4:	-0.930427	- Erro no sistema	4:	3.46945e-017
5:	0.72998	- Erro no sistema	5:	6.93889e-018
6:	0.433997	- Erro no sistema	6:	0
7:	0.0295553	- Erro no sistema	7:	0

Tempo de Execucao: 0 Segundos.

11 Variáveis

1:	3.32453	- Erro no sistema	1:	-4.16334e-017
2:	2.73364	- Erro no sistema	2:	-6.93889e-017
3:	6.42625	- Erro no sistema	3:	1.52656e-016
4:	-7.10104	- Erro no sistema	4:	-5.55112e-017
5:	3.71026	- Erro no sistema	5:	3.1225e-017
6:	-5.34167	- Erro no sistema	6:	-5.55112e-017
7:	2.13089	- Erro no sistema	7:	6.93889e-018
8:	-6.74821	- Erro no sistema	8:	2.77556e-017
9:	-1.60379	- Erro no sistema	9:	8.32667e-017
10:	-0.116113	- Erro no sistema	10:	2.08167e-017
11:	1.48304	- Erro no sistema	11:	-6.93889e-018

Tempo de Execucao: 0 Segundos.

20 Variáveis

1:	-0.926867	- Erro no sistema	1:	-6.245e-017
2:	-0.0197033	- Erro no sistema	2:	-1.00614e-016
3:	-1.44692	- Erro no sistema	3:	4.85723e-017
4:	-0.477862	- Erro no sistema	4:	-4.85723e-017
5:	-0.762897	- Erro no sistema	5:	-3.29597e-017
6:	-0.223784	- Erro no sistema	6:	-1.38778e-016
7:	-1.38855	- Erro no sistema	7:	-6.93889e-018
8:	-0.149018	- Erro no sistema	8:	-4.85723e-017
9:	0.0268548	- Erro no sistema	9:	-7.97973e-017
10:	-1.23998	- Erro no sistema	10:	-1.38778e-017
11:	0.828211	- Erro no sistema	11:	-9.71445e-017
12:	-1.20795	- Erro no sistema	12:	3.46945e-017
13:	-2.40619	- Erro no sistema	13:	-3.46945e-017
14:	0.441674	- Erro no sistema	14:	-4.16334e-017
15:	1.61413	- Erro no sistema	15:	-6.76542e-017
16:	1.46547	- Erro no sistema	16:	-2.01228e-016
17:	0.102781	- Erro no sistema	17:	-1.38778e-017
18:	3.35144	- Erro no sistema	18:	-2.63678e-016
19:	3.25018	- Erro no sistema	19:	-6.93889e-017
20:	1.3063	- Erro no sistema	20:	-1.45717e-016

Tempo de Execucao: 0 Segundos.