

**FATEC ZONA SUL**

Matheus Lopes Lourenço

Relatório de Aula - Exercícios de Estrutura de Dados

Banco de Dados Relacional

Fatec Zona Sul – DSM – 2ºSemestre

Matheus Lopes Lourenço

2ºSemestre

**Estrutura de Dados**

**Fatec Zona Sul – DSM – 2ºSemestre**

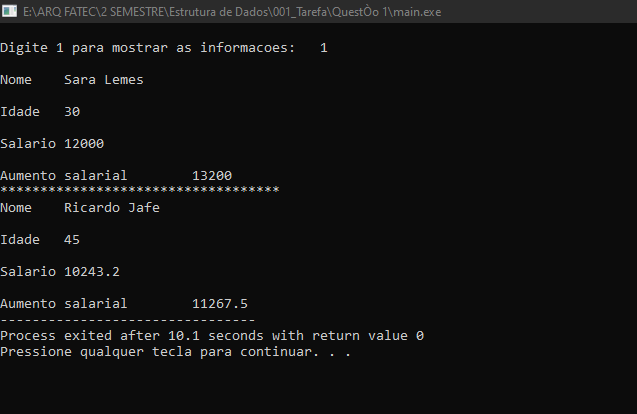
**Matheus Lopes Lourenço**

“Fazer o programa fonte de um algoritmo para exibir os vetores: String nome [] = {“Sara Lemes”, “Ricardo Jafé”}, Double salário [] = {12000, 10243.20} e Int idade [] = {30, 45}; O salário deverá ser exibido com 10% de aumento. Implemente um menu de execução.”

**1ºManeira**

**Texto

Descrição gerada automaticamente**

**2ºManeira**

**Script em C++ da 1ºManeira**

#include <iostream>

#include <string>

using namespace std;

int main() {

setlocale(LC\_ALL,"Portuguese-brasilian");

string nome1 [] = ("Sara Lemes");

string nome2 [] = ("Ricardo Jafe");

double salario [] = {12000.00, 10243.20};

int idade[]={30, 45};

cout<<("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\t");

cout<<("\nNome\t") <<nome1[0]<<("\n\t");

cout<<("\nIdade\t") <<idade[0]<<("\n\t");

cout<<("\nSalario\t") <<salario[0]<<("\n\t");

cout<<("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\t");

cout<<("\nNome\t") <<nome2[0]<<("\n\t");

cout<<("\nIdade\t") <<idade[1]<<("\n\t");

cout<<("\nSalario\t") <<salario[1]<<("\n\t");

double Aumento\_1 = salario[0]\*1.10;

double Aumento\_2 = salario[1]\*1.10;

cout<<("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\t");

cout<<("\nAumento salarial\t")<<Aumento\_1;

cout<<("\nAumento salarial\t")<<Aumento\_2<<("\n");

cout<<("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\t");

return 0;

}

**Script em C++ da 2ºManeira**

#include <iostream>

#include <string>

using namespace std;

int main() {

setlocale(LC\_ALL,"Portuguese-brasilian");

string nome1 [] = ("Sara Lemes");

string nome2 [] = ("Ricardo Jafe");

double salario [] = {12000.00, 10243.20};

int idade[]={30, 45};

char Op;

double Aumento\_1 = salario[0]\*1.10;

double Aumento\_2 = salario[1]\*1.10;

cout<<("\nDigite 1 para mostrar as informacoes: \t");

cin>>Op;

switch(Op) {

case '1' :

string nome1 [] = ("Sara Lemes");

cout<<("\nNome\t") <<nome1[0]<<("\n\t");

cout<<("\nIdade\t") <<idade[0]<<("\n\t");

cout<<("\nSalario\t") <<salario[0]<<("\n\t");

cout<<("\nAumento salarial\t")<<Aumento\_1;

cout<<("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\t");

cout<<("\nNome\t") <<nome2[0]<<("\n\t");

cout<<("\nIdade\t") <<idade[1]<<("\n\t");

cout<<("\nSalario\t") <<salario[1]<<("\n\t");

cout<<("\nAumento salarial\t")<<Aumento\_2;

break;

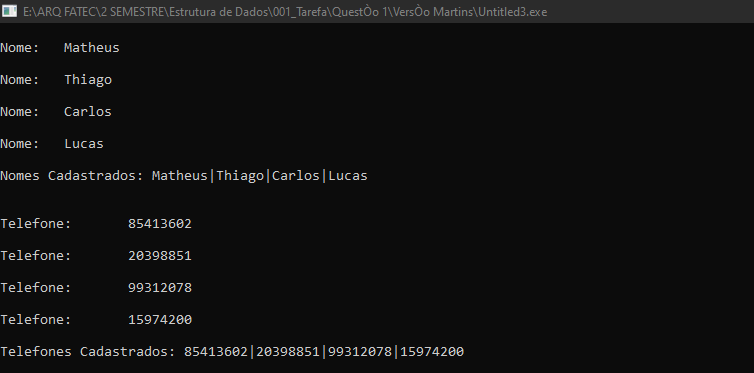
}

return 0;

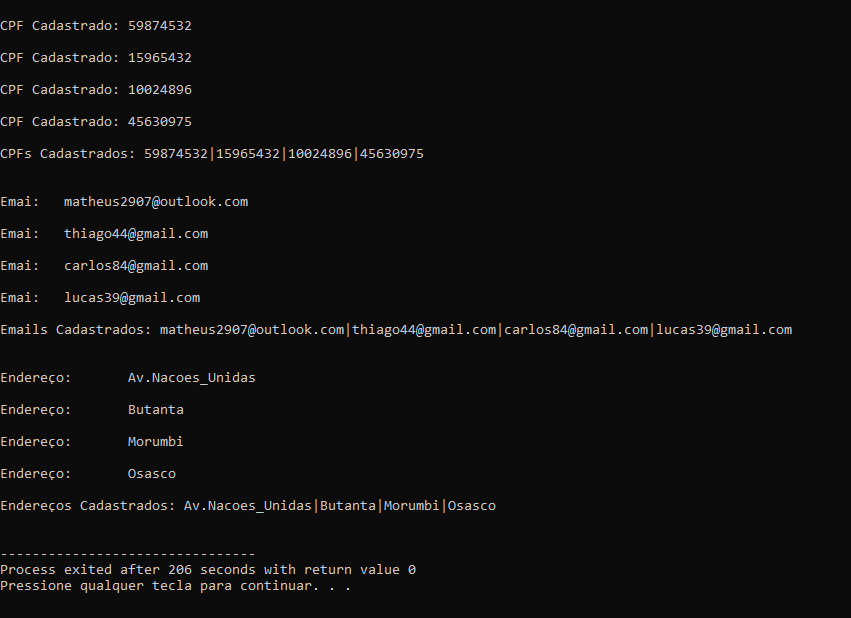
}

“Fazer um programa que leia através do teclado dados de clientes para um cadastro de um site qualquer, os dados deverão ser armazenados em vetores: nome, cpf, endereço, email e telefone. Permitir a leitura de no máximo 4 registros apenas”

**1º**



**2º**



**Script em C++**

#include "iostream"

using namespace std;

int const x=4;

string Nome[4];

int Telefone [4];

string Endereco [x];

string Email [x];

int Cpf [x];

int main() {

setlocale(LC\_ALL, "Portuguese-brazilian");

for(int i=0; i<4; i++){

cout<<"\nNome:\t"<<Nome[i];

cin>>Nome[i];

}

cout << "\nNomes Cadastrados: " << Nome [0] <<"|" << Nome [1]<<"|" << Nome [2]<<"|" << Nome [3] << "\n" << endl;

for(int i=0; i<4; i++){

cout<<"\nTelefone:\t";

cin>>Telefone[i];

}

cout << "\nTelefones Cadastrados: " << Telefone [0] <<"|" << Telefone [1]<<"|" << Telefone [2]<<"|" << Telefone [3] << "\n" << endl;

for(int i=0; i<x; i++){

cout<<"\nCPF Cadastrado: ";

cin>>Cpf[i];

}

cout << "\nCPFs Cadastrados: " << Cpf [0] <<"|" << Cpf [1]<<"|" << Cpf [2]<<"|" << Cpf [3] << "\n" << endl;

for(int i=0; i<x; i++){

cout<<"\nEmai:\t";

cin>>Email[i];

}

cout << "\nEmails Cadastrados: " << Email [0] <<"|" << Email [1]<<"|" << Email [2]<<"|" << Email [3] << "\n" << endl;

for(int i=0; i<x; i++){

cout <<"\nEndereço:\t";

cin>>Endereco[i];

}

cout << "\nEndereços Cadastrados: " << Endereco [0] <<"|" << Endereco [1]<<"|" << Endereco [2]<<"|" << Endereco [3] << "\n" << endl;

return 0;

}