

Respostas dos Exercícios de Arranjos

Mauricio Ramos Ribeiro
mauricio.ribeiro@outlook.com.br

20 de setembro de 2020

Questão 1:

```
#include <iostream>
#include <vector>

using namespace std;

int main(int argc, char *argv[]) {
    const int max = 5; // Quantidade de numeros lidos
    int num,i;
    vector<int> lista; // int lista[5];
    bool gui = true; // Controla a exibicao ou nao da saida em texto detalhada

    if (argv[2]==0) //caso argumento linha de comando == 0, desabilita exibicao detalhada
        gui = false;

    for (i=0; i<max; i++){
        bool existe = false;
        if (gui) { // Exibicao detalhada
            cout << "\033c"; // Limpa a tela
            cout << "\uEliminacao de Duplicatas" << endl;
            cout << "\u-----" << endl << endl;
            cout << "\u\u\u\u\uLista Atual:\u{";
            for (int elem : lista) cout << "\u" << elem;
            cout << "\u}";
            cout << endl << "\u\u\u\uItens da Lista:\u" << lista.size();
            cout << "\u\u\u\u\u\uItens digitados:\u" << i << endl;
        }
        cout << endl << "\u\u\u\u\uDigite um numero no intervalo [10,100]:\u";
        cin >> num;
        for (int elem : lista){ //for each
            if (elem == num) existe = true;
        }
        if (num>=10 && num <=100 && !existe) lista.push_back(num); // push_bach(x)
        acrescenta (num) no final do vetor
    }

    cout << "\u\u\u\u\u-----";
    cout << endl << "\u\u\u\u\uItens da Lista:\u" << lista.size();
    cout << "\u\u\u\u\u\uItens digitados:\u" << i << endl;
    cout << endl << "\u\u\u\u\uLista com valores unicos:\u{";
    for (int j = 0; j < lista.size(); j++){
        cout << lista[j] << "\u";
    }
    cout << "\u}" << endl << endl;
}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: dupelimitate
Eliminacao de Duplicatas
-----

Lista Atual: { }
Itens da Lista: 0      Itens digitados: 0

Digite um numero no intervalo [10,100]:
```

Figura 1: Código da Questão 1 - Parte 1

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: dupelimitate
Eliminacao de Duplicatas
-----

Lista Atual: { 10 20 }
Itens da Lista: 2      Itens digitados: 2

Digite um numero no intervalo [10,100]: 20
```

Figura 2: Código da Questão 1 - Parte 2

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash
Eliminacao de Duplicatas
-----

Lista Atual: { 10 20 30 }
Itens da Lista: 3      Itens digitados: 4

Digite um numero no intervalo [10,100]: 40
-----
Lista com valores unicos: { 10 20 30 40 }

mrribeiro@DESKTOP-EJPO918:~/estudos/poo2020/POO2020N/S04/1.dupelimitate/build$
```

Figura 3: Código da Questão 1 - Parte 3

Questão 2:

```
#include <iostream>

using namespace std;

int main(int argc, char *argv[]) {
    int num;

    cout << "\033c"; // Limpar tela
    if (argc == 2) {
        num = atoi(argv[1]);
    }
    else {
        cout << endl << " Digite o numero de elementos do arranjo: ";
        cin >> num;
    }

    int lista[num];
    for (int i=0; i<num; i++){
        lista[i] = 0;
    }

    cout << endl << "Indice   Valor" << endl;
    cout << "-----" << endl;

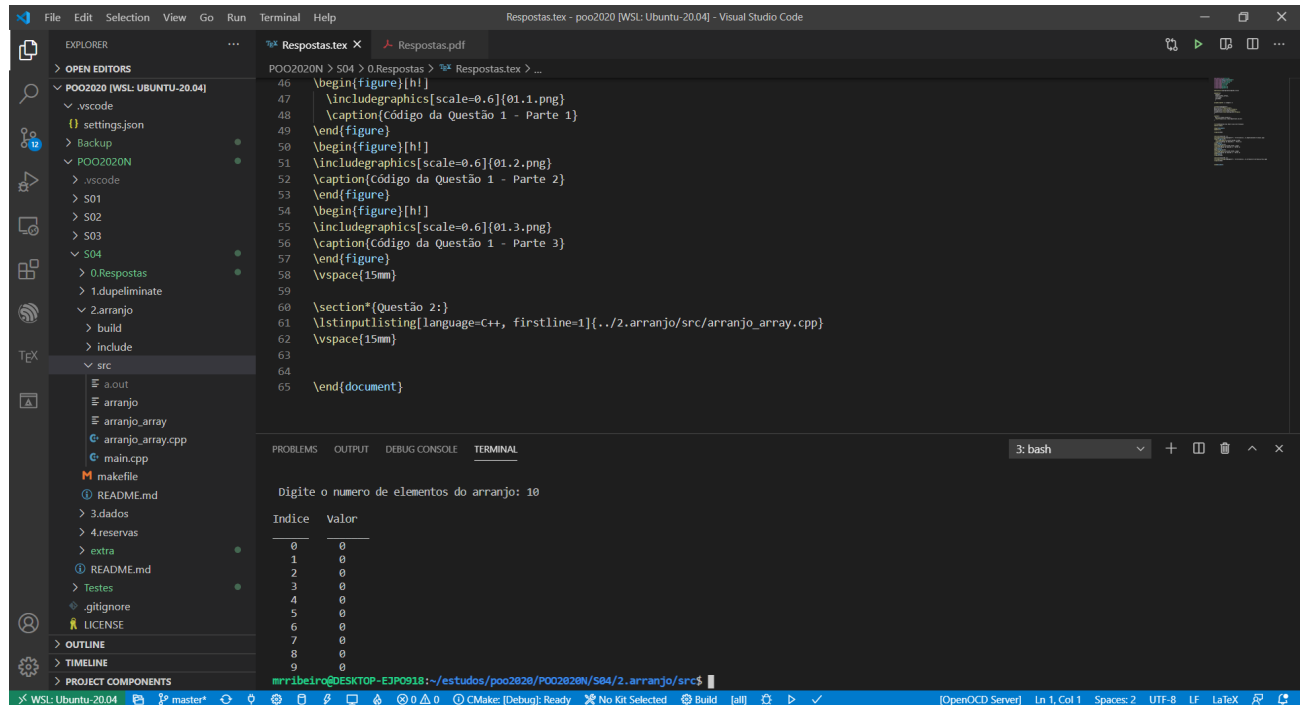
    for (int i=0; i<num;i++){
```

```

        lista[i];
        cout << "uuu" << i << "uuuuuuu" << lista[i] << endl;
    }

}

```



Questão 3:

```

#include <iostream>
#include <iomanip>

using namespace std;

int main(void){
    const int numeroDeLancamentos = 36000;
    int d1,d2;
    int vezes[36] = {
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0 };
    // Eu usaria vezes[6][6] = { { 1,2,3,4,5,6}
    //                               { 1,2,3,4,5,6}}

    //Geracao dos Dados
    cout << endl;
    for (int i=0; i<numeroDeLancamentos; i++){
        d1 = 1 + rand() % 6;
        d2 = 1 + rand() % 6;
    }
}

```

```

// cout << d1 << " e " << d2 << " = " << d1+d2 << " | " ; *Impressao dos resultados d
for(int j=0;j<6;j++){
    if (d1==j+1){
        for(int i=0;i<6;i++){
            if(d2==i+1)
                vezes[i+j*6] += 1;
        }
    }
}

//Apresentacao Na Tela
cout << endl << endl;
cout << "uuuuu1uuuu2uuuu3uuuu4uuuu5uuuu6" << endl;
cout << "-----" << endl;
for(int j=0;j<6;j++){
    cout << j+1 << " | ";
    for(int i=0;i<6;i++){
        cout << right << setw(4) << vezes[i+j*6] << " ";
    }
    cout << endl;
}
cout << endl;
}

```

The screenshot shows a Visual Studio Code editor with a LaTeX document titled 'Respostas.tex'. The document contains code for generating a grid of numbers. The terminal output shows a 6x6 grid of numbers:

```

1  989 1035 1035 1024 998 1020
2  974 1014 1029 994 953 970
3 1003 1041 1056 999 1036 1042
4  997 997 1057 1015 988 1003
5  938 961 950 969 1030 958
6  975 926 1049 1033 1007 935

```

Figura 5: Código da Questão 3

Questão 4:

```

#include <iostream>
#include <iomanip>

```

```

using namespace std;

string LIMPATELA = "\033c";
const int NUMEROASSENTOS = 10;           // Numero total de assentos
// Numero que indica a partir de qual assento a classe e economica iniciando de 0
// Exemplo: Iniciando na poltrona 6, colocar 5
const int INICIOCLASSEECONOMICA = 5;

void ImprimeCabecalho(void);
void ImprimeCabecalho(string nome);
void ImprimeCabecalho(string nome, int classe);

void ImprimeAssentosDisponiveis(bool assentosOcupados[], int classe);

int main(void){

    string passageiros[NUMEROASSENTOS] {};
    // bool assentos[numeroAssentos] {0,0,0,0,0,0,0,0,0,0}; poderia ser utilizado.
    // mas perderiamos o redimensionamento no numero de assentos
    bool assentosOcupados[NUMEROASSENTOS] {};    for(auto elem : assentosOcupados) elem = false;
    string nome[NUMEROASSENTOS] {};
    string nomeAux;
    int assentoAux;
    int classe;
    int digita = 1;
    int cheio;
    bool sv1=false,sv2=false;
    bool troca;
    int temp;

    while (digita)
    {
        sv1=false;sv2=false;

        //Entrada do nome do passageiro
        ImprimeCabecalho();

        cout << "\nDigite seu nome completo: ";
        //getline(cin, nomeAux);
        cin >> nomeAux;

        //Entrada da classe
        ImprimeCabecalho(nomeAux);
        cout << "\nDigite 1 para primeira classe ou 2 para classe economica: ";
        cin >> classe;
        while (classe < 1 || classe > 2){
            ImprimeCabecalho(nomeAux);
            cout << "\nClasse Invalida. Digite Novamente!" << endl;
            cout << "\nDigite 1 para primeira classe ou 2 para classe economica: ";
            cin >> classe;
        }

        if(classe==1){
            for (int i=0; i<INICIOCLASSEECONOMICA;i++){
                if(assentosOcupados[i])
                    sv1 = true;
            }
        }
        if(classe==2){
            for (int i=INICIOCLASSEECONOMICA; i<NUMEROASSENTOS;i++){
                if(assentosOcupados[i])
                    sv2 = true;
            }
        }
    }
}

```

```

    }
}

if(sv1){
    cout << "Nao existem mais assentos na classe escolhida!" << endl;
    cout << "Aceita trocar para a Classe Economica? (1-sim/0-Nao) ";
    cin >> troca;
    if (troca)
        classe = 2;
    else {
        cout << "0 proximo voo parte em 3 horas" << endl;
        cout << "Digite 1 para continuar ou 0 para encerrar o programa: ";
        cin >> temp;
        if (temp)
            continue;
        else break;
    }
}

if(sv2){
    cout << "Nao existem mais assentos na classe escolhida!" << endl;
    cout << "Aceita trocar para a Primeira Classe? (1-sim/0-Nao) ";
    cin >> troca;
    if (troca)
        classe = 1;
    else {
        cout << "0 proximo voo parte em 3 horas" << endl;
        cout << "Digite 1 para continuar ou 0 para encerrar o programa: ";
        cin >> temp;
        if (temp)
            continue;
        else break;
    }
}

//Entrada do assento
if (classe == 1){
    ImprimeCabecalho(nomeAux, classe);
    ImprimeAssentosDisponiveis(assentosOcupados, classe);
    cout << endl << "Qual assento deseja? ";
    cin >> assentoAux;
    while (assentosOcupados[assentoAux-1] || assentoAux < 1 || assentoAux > INICIOCLASSEECONOMICA)
        ImprimeCabecalho(nomeAux, classe);
    cout << "Assento ocupado ou invalido" << endl;
    ImprimeAssentosDisponiveis(assentosOcupados, classe);
    cout << endl << "Digite outro assento: ";
    cin >> assentoAux;
}

else if (classe == 2){
    ImprimeCabecalho(nomeAux, classe);
    ImprimeAssentosDisponiveis(assentosOcupados, classe);
    cout << endl << "Qual assento deseja? ";
    cin >> assentoAux;
    while (assentosOcupados[assentoAux-1] || assentoAux < INICIOCLASSEECONOMICA || assentoAux > INICIOCLASSEECONOMICA)
        ImprimeCabecalho(nomeAux, classe);
    cout << "Assento ocupado ou invalido" << endl;
    ImprimeAssentosDisponiveis(assentosOcupados, classe);
    cout << endl << "Digite outro assento: ";
    cin >> assentoAux;
}

}

//Entra dados no array
nome[assentoAux-1] = nomeAux;

```

```

        assentosOcupados[assentoAux-1] = true;

        //Imprime dados passageiro
        ImprimeCabecalho();
        cout << "Passageiro:" << nome[assentoAux-1] << endl;
        cout << "Poltrona:   " << assentoAux << endl;
        cout << "Classe:   " << ((classe==1)? "PrimeiraClasse.": "ClasseEconomica.");

        cout << endl << endl << "Opcao: (1-sim/0-Nao): ";
        cin >> digita;

        nomeAux = "";
        assentoAux = 0;
        cin.clear();
        cout.flush();
        fflush(stdin);

    }

    // Imprime dados das reservas e sai
    ImprimeCabecalho();
    cout << "Resumo do Voo:" << endl;
    cout << "-----" << endl;
    for (int i = 0; i<NUMEROASSENTOS; i++){
        cout << "Poltrona:   " << setw(2) << i+1 << endl;
        cout << "Passageiro:  " << nome[i] << endl;
        cout << "Classe:   " << ((i<INICIOCLASSEECONOMICA)? "PrimeiraClasse.": "ClasseE"
    }

}

void ImprimeCabecalho(string nome){
    cout << LIMPATELA;
    cout << endl << "Sistema de reservas" << endl;
    cout << "-----" << endl;
    if (nome != ""){
        cout << "Passageiro:" << nome << endl << endl;
    }
}

void ImprimeCabecalho(string nome, int classe){
    cout << LIMPATELA;
    cout << endl << "Sistema de reservas" << endl;
    cout << "-----" << endl;
    if (nome != ""){
        cout << ((classe==1)? "PrimeiraClasse.": "ClasseEconomica.");
        cout << "Passageiro:" << nome << endl << endl;
    }
}

void ImprimeCabecalho(void){
    cout << LIMPATELA;
    cout << endl << "Sistema de reservas" << endl;
    cout << "-----" << endl << endl;
}

void ImprimeAssentosDisponiveis(bool assentosOcupados[], int classe){
    if (classe == 1){
        cout << "Assentos disponiveis na PrimeiraClasse:";
        for (int i = 0; i < INICIOCLASSEECONOMICA; i++){
            if (!assentosOcupados[i])
                cout << i+1 << " ";
        }
    }
}

```

```

else if(classe == 2){
    cout << "\nAssentos disponiveis na Classe Economica:\n";
    for (int i = INICIOCLASSEECONOMICA; i < NUMEROASSENTOS; i++){
        if (!assentosOcupados[i])
            cout << i+1 << "\n";
    }
}
}
}

```

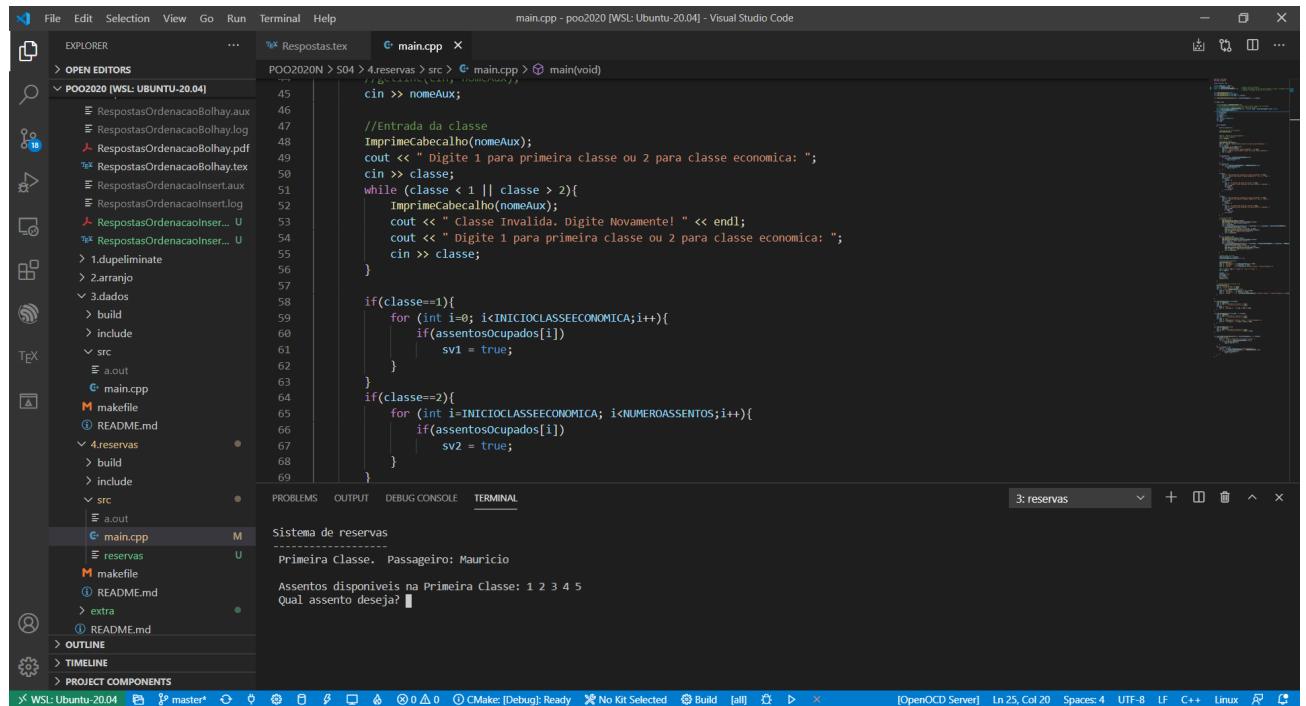


Figura 6: Código da Questão 4 - Parte 1

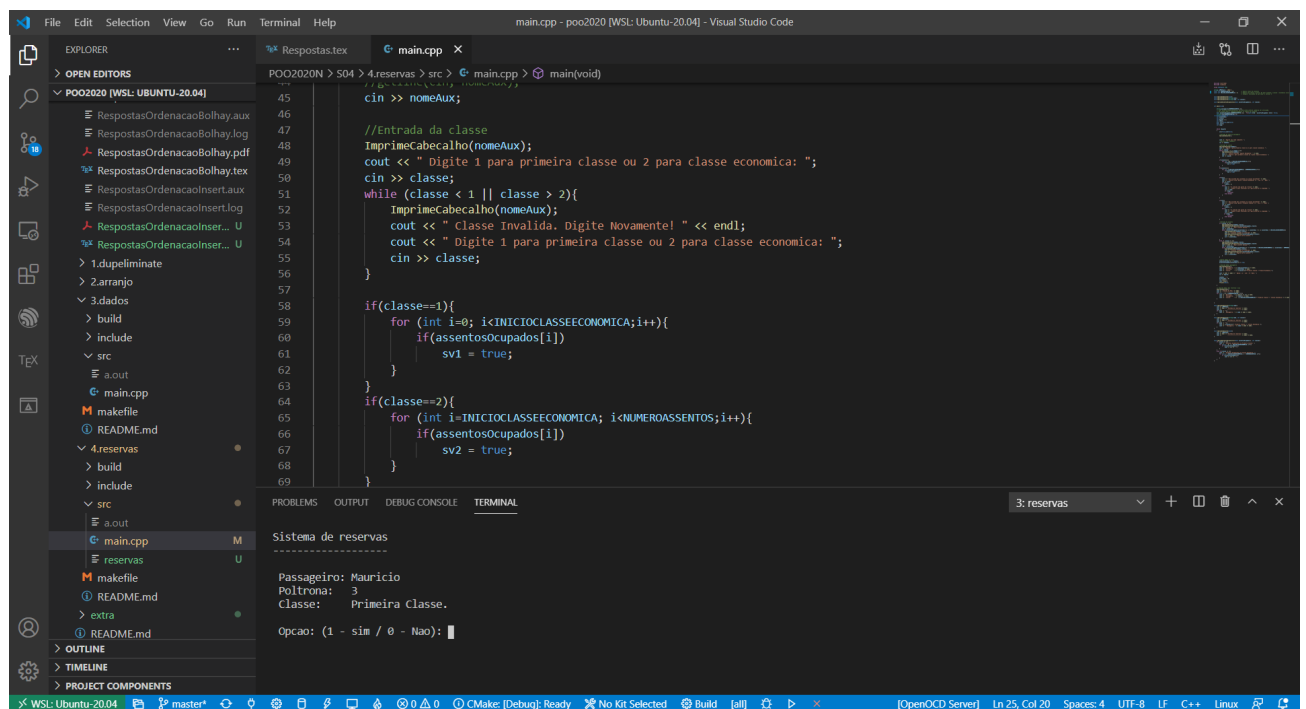


Figura 7: Código da Questão 4 - Parte 2


```

45  cin >> nomeAux;
46
47  //Entrada da classe
48  ImprimeCabecalho(nomeAux);
49  cout << " Digite 1 para primeira classe ou 2 para classe economica: ";
50  cin >> classe;
51  while (classe < 1 || classe > 2){
52      ImprimeCabecalho(nomeAux);
53      cout << " Classe Invalida. Digite Novamente! " << endl;
54      cout << " Digite 1 para primeira classe ou 2 para classe economica: ";
55      cin >> classe;
56  }
57
58  if(classe==1){
59      for (int i=0; i<INICIOCLASSEECONOMICA;i++){
60          if(assentosOcupados[i])
61              sv1 = true;
62      }
63  }
64  if(classe==2){
65      for (int i=INICIOCLASSEECONOMICA; i<NUMEROASSENTOS;i++){
66          if(assentosOcupados[i])
67              sv2 = true;
68      }
69  }

```

Terminal Output:

```

Sistema de reservas
Passageiro: Adriana
Digite 1 para primeira classe ou 2 para classe economica: 1
Nao existem mais assentos na classe escolhida!
Aceita trocar para a Classe Economica? (1 - sim / 0 - Nao)

```

Figura 8: Código da Questão 4 - Parte 3

```

45  cin >> nomeAux;
46
47  //Entrada da classe
48  ImprimeCabecalho(nomeAux);
49  cout << " Digite 1 para primeira classe ou 2 para classe economica: ";
50  cin >> classe;
51  while (classe < 1 || classe > 2){
52      ImprimeCabecalho(nomeAux);
53      cout << " Classe Invalida. Digite Novamente! " << endl;
54      cout << " Digite 1 para primeira classe ou 2 para classe economica: ";
55      cin >> classe;
56  }
57
58  if(classe==1){
59      for (int i=0; i<INICIOCLASSEECONOMICA;i++){
60          if(assentosOcupados[i])
61              sv1 = true;
62      }
63  }
64  if(classe==2){
65      for (int i=INICIOCLASSEECONOMICA; i<NUMEROASSENTOS;i++){
66          if(assentosOcupados[i])
67              sv2 = true;
68      }
69  }

```

Terminal Output:

```

Sistema de reservas
-----
Classe Economica. Passageiro: Adriana
Assentos disponiveis na Classe Economica: 6 7 8 9 10
Qual assento deseja?

```

Figura 9: Código da Questão 4 - Parte 4

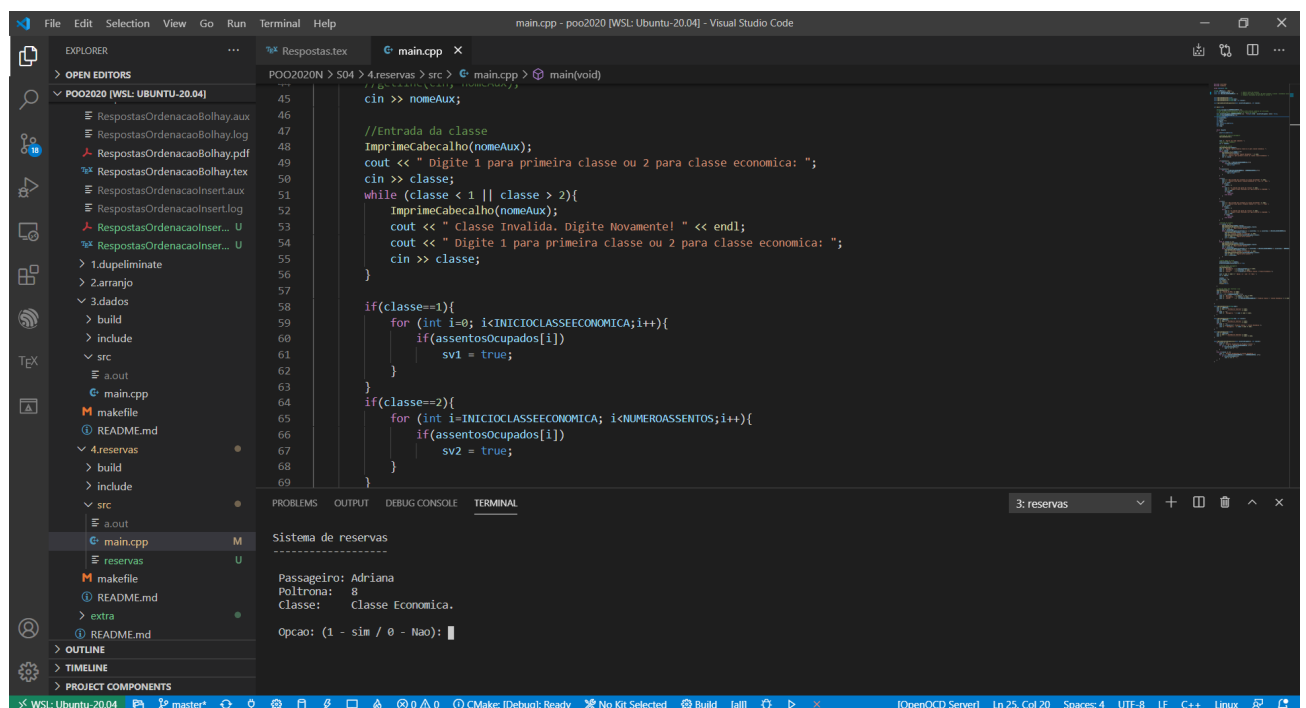


Figura 10: Código da Questão 4 - Parte 5