

Universidade Estadual do Rio Grande do Sul - UERGS

Programação Orientada a Objetos - 2020/01

Professor André Borin

Grupo 4 - Listagem de Rotas

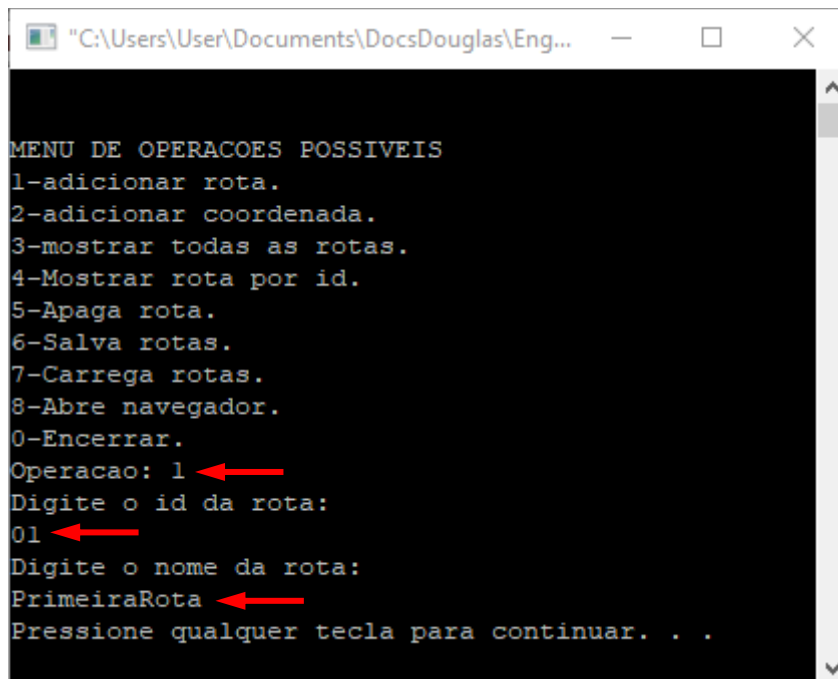
Douglas Alves da Cruz

Gabriel Fachinello

Lucas Silva Ennes

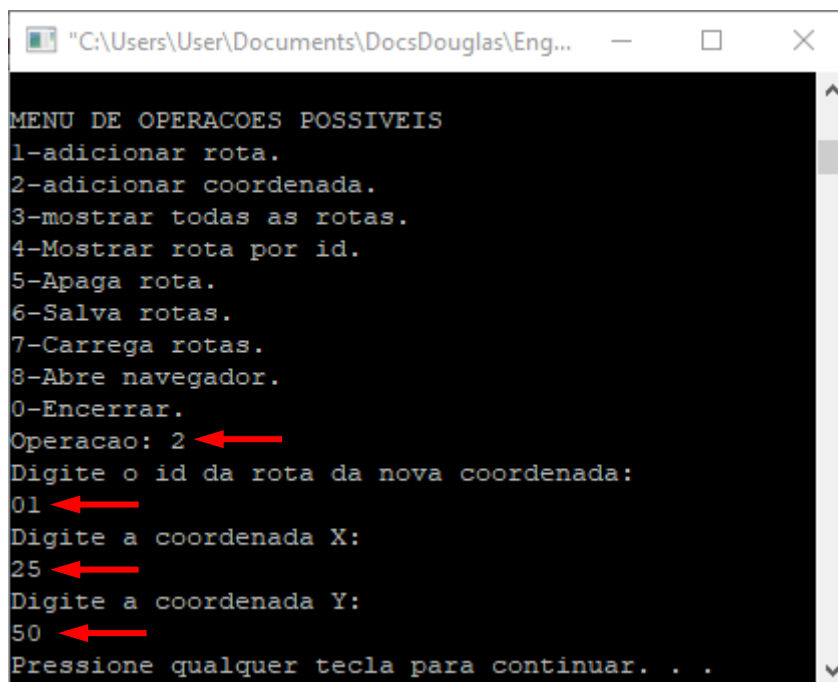
Larissa Justen

Interface de Operação



```
"C:\Users\User\Documents\DocsDouglas\Eng..."  
  
MENU DE OPERACOES POSSIVEIS  
1-adicionar rota.  
2-adicionar coordenada.  
3-mostrar todas as rotas.  
4-Mostrar rota por id.  
5-Apaga rota.  
6-Salva rotas.  
7-Carrega rotas.  
8-Abre navegador.  
0-Encerrar.  
Operacao: 1  
Digite o id da rota:  
01  
Digite o nome da rota:  
PrimeiraRota  
Pressione qualquer tecla para continuar. . .
```

Imagem: 1-Adicionar Rota



```
"C:\Users\User\Documents\DocsDouglas\Eng..."  
  
MENU DE OPERACOES POSSIVEIS  
1-adicionar rota.  
2-adicionar coordenada.  
3-mostrar todas as rotas.  
4-Mostrar rota por id.  
5-Apaga rota.  
6-Salva rotas.  
7-Carrega rotas.  
8-Abre navegador.  
0-Encerrar.  
Operacao: 2  
Digite o id da rota da nova coordenada:  
01  
Digite a coordenada X:  
25  
Digite a coordenada Y:  
50  
Pressione qualquer tecla para continuar. . .
```

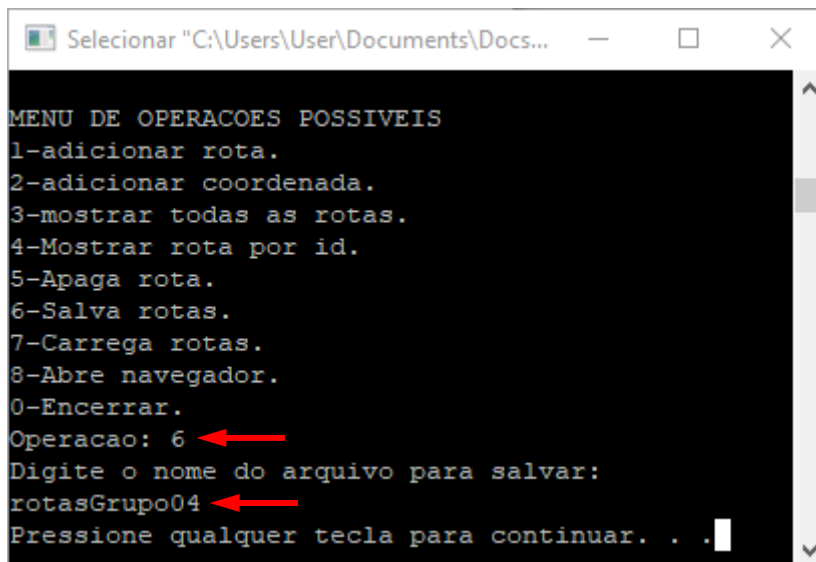
Imagem: 2-Adicionar Coordenada

```
"C:\Users\User\Documents\DocsDouglas\Eng...  —  □  ×  
  
MENU DE OPERACOES POSSIVEIS  
1-adicionar rota.  
2-adicionar coordenada.  
3-mostrar todas as rotas.  
4-Mostrar rota por id.  
5-Apaga rota.  
6-Salva rotas.  
7-Carrega rotas.  
8-Abre navegador.  
0-Encerrar.  
Operacao: 3  
Todos items do objeto rotas:  
id: 1, nome: PrimeiraRota, rotas:  
Pressione qualquer tecla para continuar. . .
```

Imagem 3: Mostrar todas as Rotas

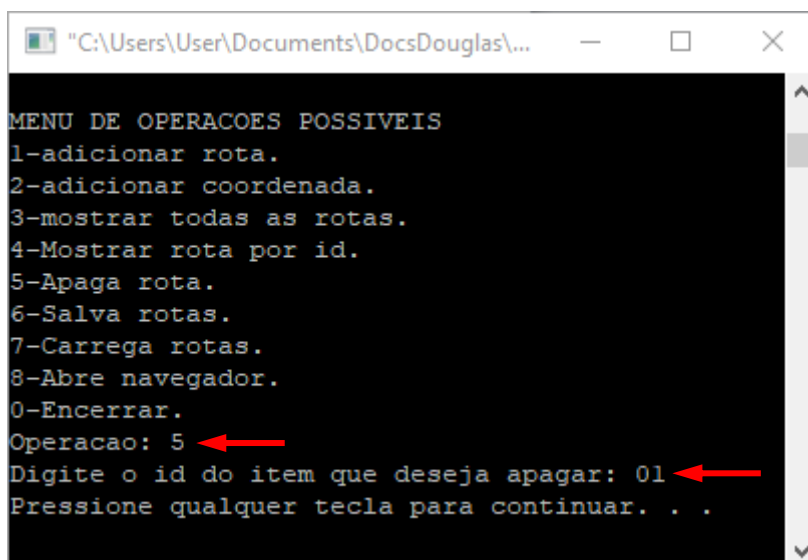
```
"C:\Users\User\Documents\DocsDouglas\Eng...  —  □  ×  
  
MENU DE OPERACOES POSSIVEIS  
1-adicionar rota.  
2-adicionar coordenada.  
3-mostrar todas as rotas.  
4-Mostrar rota por id.  
5-Apaga rota.  
6-Salva rotas.  
7-Carrega rotas.  
8-Abre navegador.  
0-Encerrar.  
Operacao: 4  
Digite o id da rota: 01  
id: 1, nome: PrimeiraRota, rotas:  
Pressione qualquer tecla para continuar. . .
```

Imagem4: Mostrar Rota por Id



```
MENU DE OPERACOES POSSIVEIS
1-adicionar rota.
2-adicionar coordenada.
3-mostrar todas as rotas.
4-Mostrar rota por id.
5-Apaga rota.
6-Salva rotas.
7-Carrega rotas.
8-Abre navegador.
0-Encerrar.
Operacao: 6
Digite o nome do arquivo para salvar:
rotasGrupo04
Pressione qualquer tecla para continuar. . .
```

Imagem 5: Salva Rotas



```
MENU DE OPERACOES POSSIVEIS
1-adicionar rota.
2-adicionar coordenada.
3-mostrar todas as rotas.
4-Mostrar rota por id.
5-Apaga rota.
6-Salva rotas.
7-Carrega rotas.
8-Abre navegador.
0-Encerrar.
Operacao: 5
Digite o id do item que deseja apagar: 01
Pressione qualquer tecla para continuar. . .
```

Imagem 6: Apaga Rota

```
"C:\Users\User\Documents\DocsDouglas\E...  -  □  X

MENU DE OPERACOES POSSIVEIS
1-adicionar rota.
2-adicionar coordenada.
3-mostrar todas as rotas.
4-Mostrar rota por id.
5-Apaga rota.
6-Salva rotas.
7-Carrega rotas.
8-Abre navegador.
0-Encerrar.
Operacao: 7
Digite o nome do arquivo para carregar:
rotasGrupo04
Pressione qualquer tecla para continuar. . .

MENU DE OPERACOES POSSIVEIS
1-adicionar rota.
2-adicionar coordenada.
3-mostrar todas as rotas.
4-Mostrar rota por id.
5-Apaga rota.
6-Salva rotas.
7-Carrega rotas.
8-Abre navegador.
0-Encerrar.
Operacao: 3
Todos items do objeto rotas:
id: 1, nome: PrimeiraRota, rotas:
Pressione qualquer tecla para continuar. . .
```

Imagem 7: Carrega Rotas e apresenta dados

```
"C:\Users\User\Documents\DocsDouglas\E...  -  □  X

MENU DE OPERACOES POSSIVEIS
1-adicionar rota.
2-adicionar coordenada.
3-mostrar todas as rotas.
4-Mostrar rota por id.
5-Apaga rota.
6-Salva rotas.
7-Carrega rotas.
8-Abre navegador.
0-Encerrar.
Operacao:
8
Abrindo p|igina web...
Pressione qualquer tecla para continuar. . .
```

Imagem 8: Abre Navegador

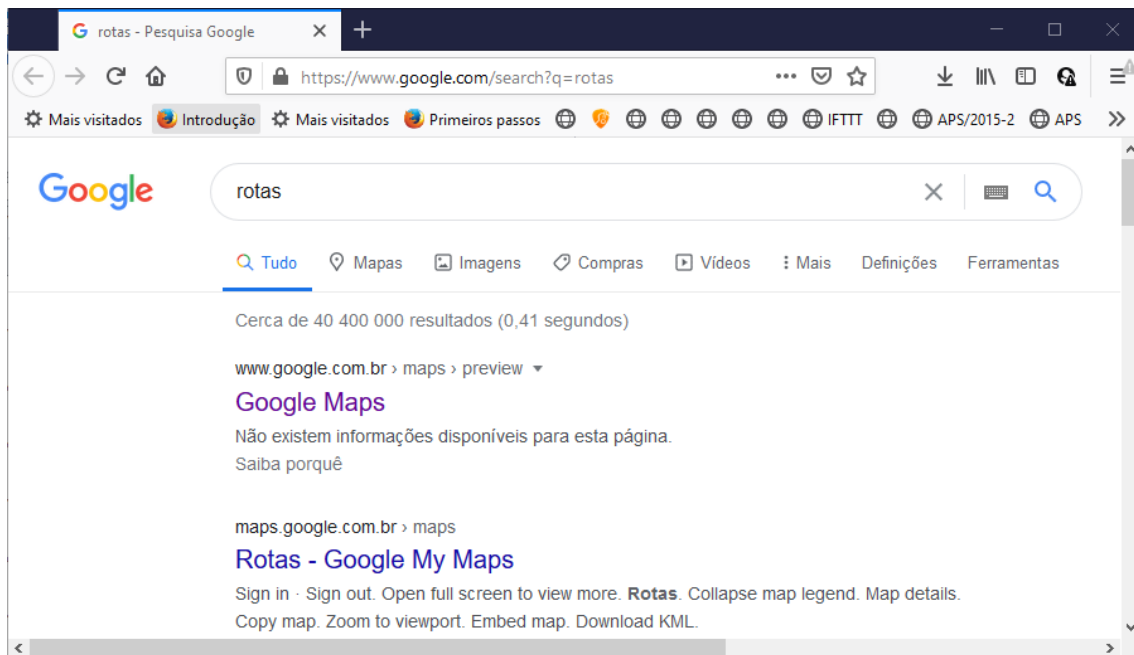


Imagem 9: resultado do comando de Abrir Navegador

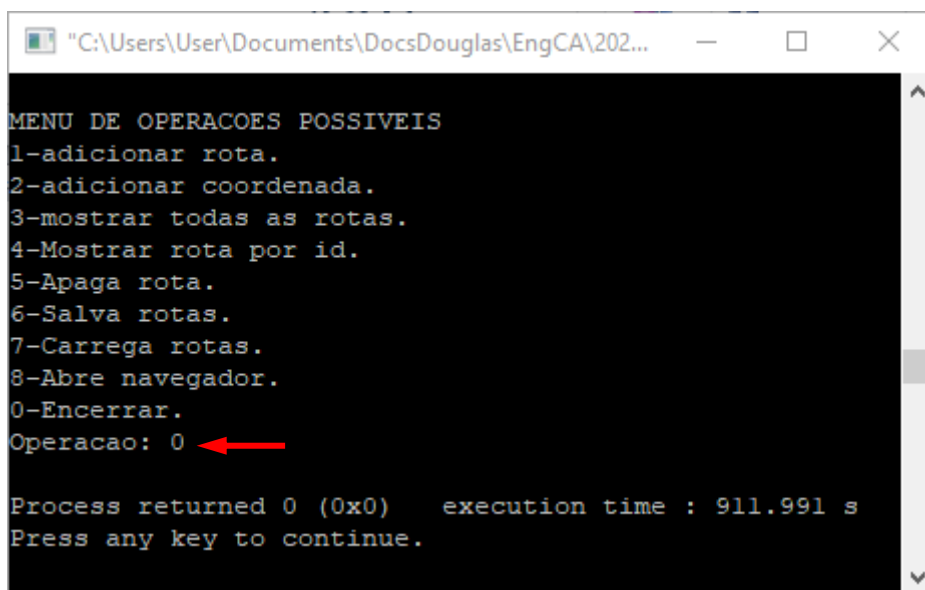
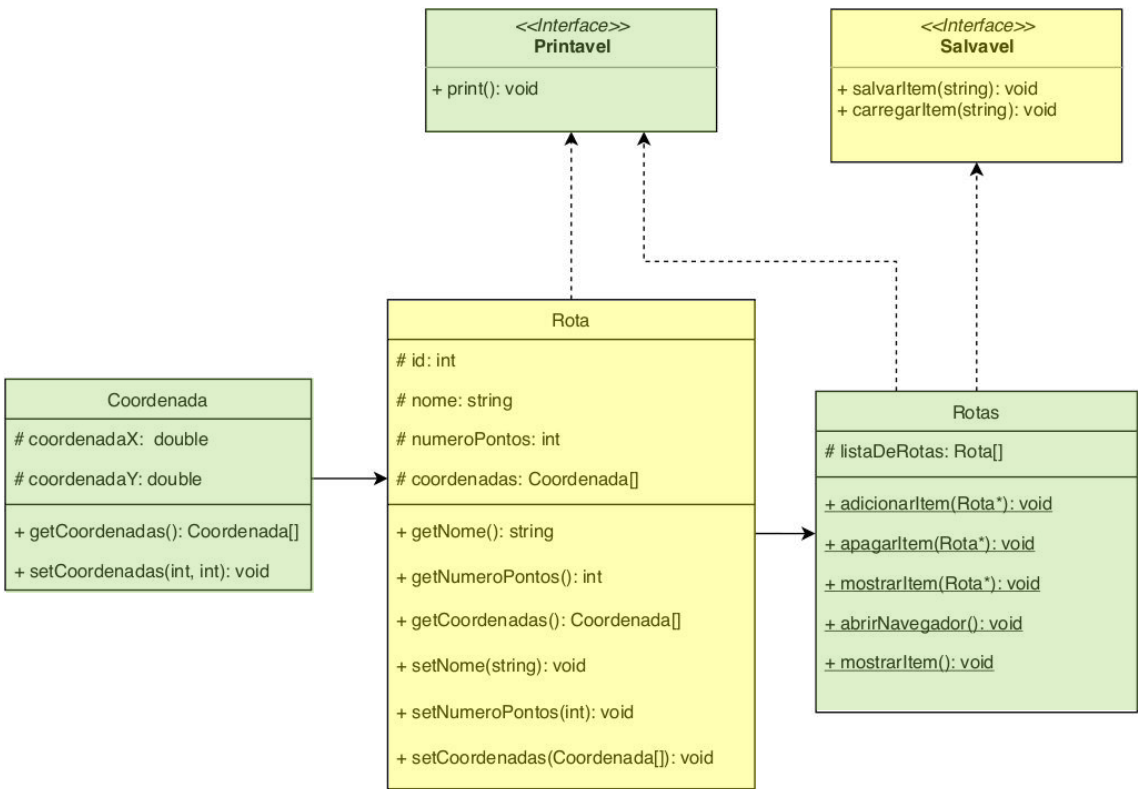


Imagem 10: Encerrar

Diagrama de Classes do Sistema de listagem de rotas:



listagem de rotas ("Rota 1", 2 (pontos), (x1,y1),(x2,y2), "Rota 2",3 pontos, ...)

Código Fonte; C++

```
#include <iostream>
#include <stdlib.h>
#include <vector>
#include <ctime>
#include <fstream>
#include <string>

using namespace std;

class Printavel {
public:
    virtual ~Printavel() {}
    virtual void print() = 0;
};

class Salvavel {
public:
    virtual ~Salvavel() {}
    virtual void salvarItem(string item) = 0;
    virtual void carregarItem(string item) = 0;
};

class Coordenada{
protected:
    double coordenadaX;
    double coordenadaY;

public:
    Coordenada(double x, double y) {
        coordenadaX = x;
        coordenadaY = y;
    }
    Coordenada() {
    }

    vector<double> getCoordenadas() {
        return { coordenadaX, coordenadaY };
    }

    void setCoordenadas(double x, double y) {
        coordenadaX = x;
        coordenadaY = y;
    }
};

class Rota : public Printavel {
protected:
    int id;
    string nome;
    vector<Coordenada> coordenadas;

public:
    Rota(){ }

    Rota(int id, string nome){ this->id = id; this->nome = nome;}

    string getNome() {
        return nome;
    }

    int getId() {
        return id;
    }

    vector<Coordenada> getCoordenadas() {
        return coordenadas;
    }

    void setNome(string nome) {
```



```

        this->nome = nome;
    }

    void setId(int id) {
        this->id = id;
    }

    void setCoordenadas(vector<Coordenada> coordenadas) {
        this->coordenadas = coordenadas;
    }

    void print() {
        cout << "id: " << this->getId() << ", nome: " << this->getNome() << ",
rotas: ";
        for(Coordenada coord : coordenadas){
            cout << "(" << coord.getCoordenadas()[0] << ", " <<
coord.getCoordenadas()[1] << ")", ";
        }
        cout << endl;
    }

    void adicionarCoordenada(Coordenada coordenada){
        coordenadas.push_back(coordenada);
    }
};

```

```

class Rotas : public Printavel, public Salvavel {
protected:
    vector<Rota> listaDeRotas;

public:
    Rotas() {}

    void adicionarItem(Rota rota) {
        listaDeRotas.push_back(rota);
    }

    void apagarItem(int id) {
        for(int i = 0; i < listaDeRotas.size(); i++){
            if(listaDeRotas.at(i).getId() == id){
                listaDeRotas.erase(listaDeRotas.begin() + i);
            }
        }
    }

    void print() {
        for(Rota item : listaDeRotas){
            item.print();
        }
    }

    void mostrarItem(int id) {
        for(int i = 0; i < listaDeRotas.size(); i++){
            if(listaDeRotas.at(i).getId() == id){
                listaDeRotas.at(i).print();
            }
        }
    }

    void abrirNavegador() {
        cout << "Abrindo página web..." << endl;
        system("\"C:\\Program Files\\Mozilla Firefox\\firefox.exe\"
google.com/search?q=rotas");
    }

    void salvarItem(string filename){
        ofstream file (filename);
        if (file.is_open())
        {
            for(Rota item : listaDeRotas){
                file << item.getId() << ',' << item.getNome();
                for(Coordenada coord : item.getCoordenadas()){
                    file << ',' << coord.getCoordenadas()[0] << ',' <<
coord.getCoordenadas()[1] << endl;
                }
            }
        }
    }
}

```

```

        file.close();
    }
    else {
        cout << "Unable to open file";
    }
}

void carregarItem(string arquivo){
    int i = 0;
    string linha;
    int id;
    double x, y;
    string nome;
    ifstream file(arquivo);
    if (file.is_open())
    {
        while( getline(file, linha) ){
            id = stoi(linha.substr(0,linha.find(',')));
            linha.erase(0,linha.find(',')+1);
            nome = linha.substr(0,linha.find(',') );
            linha.erase(0,linha.find(',')+1);
            listaDeRotas.push_back(Rota(id, nome));
            while(linha.find(',') != -1){
                x = stod(linha.substr(0,linha.find(',') ));
                linha.erase(0,linha.find(',')+1);
                y = stod(linha.substr(0,linha.find(',') ));
                linha.erase(0,linha.find(',')+1);
                listaDeRotas.back().adicionarCoordenada(Coordenada(x,y));
            }
        }
        file.close();
    }
    else {
        cout << "Não foi possivel abrir o arquivo";
    }
}

int validarId(int id){
    int existe = -1;
    for(Rota r : listaDeRotas){
        if(r.getId() == id){
            existe = 1;
            break;
        }
    }
    return existe;
}

Rota getRota(int id){
    for(Rota r : listaDeRotas){
        if(r.getId() == id){
            return r;
        }
    }
}

};

int main() {
    Rotas rotas;
    double voltaMenu;
    double x, y;
    string nome;
    int id;
    int entradaMenu, idItemSelecionado;
    string arquivo;
    Coordenada coordenada;

    while(1 == 1) {
        cout << "\n\nMENU DE OPERACOES POSSIVEIS " << endl;
        cout << "1-adicionar rota." << endl;
        cout << "2-adicionar coordenada." << endl;
        cout << "3-mostrar todas as rotas." << endl;
        cout << "4-Mostrar rota por id." << endl;
        cout << "5-Apaga rota." << endl;
    }
}

```

```

        cout << "6-Salva rotas." << endl;
        cout << "7-Carrega rotas." << endl;
        cout << "8-Abre navegador." << endl;
        cout << "0-Encerrar." << endl;
        cout << "Operacao: ";
        cin >> entradaMenu;

        if(entradaMenu == 0) break;

        switch (entradaMenu) {

            // cria rota - ok
            case 1:
                cout << "Digite o id da rota: " << endl;
                cin >> id;
                if(rotas.validarId(id) == 1){
                    cout << "o id inserido já existe!" << endl;
                    break;
                }
                cout << "Digite o nome da rota: " << endl;
                cin >> nome;
                rotas.adicionarItem(Rota(id,nome));
                system("pause");
                break;
            // cria coordenada - ok
            case 2:
                cout << "Digite o id da rota da nova coordenada: " << endl;
                cin >> x;
                if(rotas.validarId(id) == -1){
                    cout << "o id inserido não existe!" << endl;
                    break;
                }
                cout << "Digite a coordenada X: " << endl;
                cin >> x;
                cout << "Digite a coordenada Y: " << endl;
                cin >> y;
                coordenada = Coordenada(x,y);
                rotas.getRota(id).adicionarCoordenada(coordenada);
                system("pause");
                break;

            // mostra rotas - ok
            case 3:
                cout << "Todos itens do objeto rotas: " << endl;
                rotas.print();
                system("pause");
                break;

            // mostra rota - ok
            case 4:
                cout << "Digite o id da rota: ";
                cin >> id;
                if(rotas.validarId(id) == -1){
                    cout << "o id inserido não existe!" << endl;
                    break;
                }
                rotas.mostrarItem(id);
                system("pause");
                break;

            // apaga rota - ok
            case 5:
                cout << "Digite o id do item que deseja apagar: ";
                cin >> id;
                if(rotas.validarId(id) == -1){
                    cout << "o id inserido não existe!" << endl;
                    break;
                }
                rotas.apagarItem(id);
                system("pause");
                break;

            // salva lista
            case 6:
                cout << "Digite o nome do arquivo para salvar: " << endl;
                cin >> arquivo;
                rotas.salvarItem(arquivo);

```

```
        system("pause");
        break;

        // carrega lista
        case 7:
            cout << "Digite o nome do arquivo para carregar: " << endl;
            cin >> arquivo;
            rotas.carregarItem(arquivo);
            system("pause");
            break;

        // abrir navegador
        case 8:
            rotas.abrirNavegador();
            system("pause");
            break;

        default:
            cout << "Opção inválida." << endl;
            cout << endl;
            system("pause");
            break;
    }
}

return 0;
}
```