Agência de viagens

Programação orientada a objetos

Membros do grupo

Priscila Leite Dos Santos Silva

Mateus Regasi Gomes Martins

Lucas de Lima Moura

Guilherme Martins Rangel

Gustavo Gomes Viana

Classes

Csv

Funções

Cliente

Hotel

Voo

Paralelo

Csv

Csv

CSV

```
public class CSV {
public List<Cliente> leitorClientes(String arquivo)
public List<Hotel> leitorHoteis(String arquivo)
public List<Voo> leitorVoos(String arquivo)
public void escreverFinal(String dados, String arquivo)}
```

Funções

Funções

Funcoes

```
public class Funcoes {
public int melhorHotel(List<Hotel> hoteis, Cliente c)
public List<Voo> melhorCaminho(String saida,String chegada,
                               List<Voo> voos)}
```

Cliente

Cliente

Cliente (sequencial)

```
public class Cliente {
public Cliente(String nome,String saida, String chegada,int estadia,
               int estrelas,float orcamento)
public String getNome()
public String getSaida()
public String getChegada()
public double getOrcamento()
public int getEstrelas()
public int getEstadia()
public String toString()
public double verOrcamento(Hotel hotel, List<Voo> voos)
public String reservar(Hotel h, List<Voo> v)}
```

Hotel

Hotel

Hotel

```
public class Hotel {
public Hotel(String localizacao, String nome, int vagas, double preco,
             int estrelas)
public String getNome()
public String getLocalizacao()
public double getPreco()
public int getEstrelas()
public int getVagas()
public void reservar()
public String toString()}
```

Voo

Voo

Voo

```
public class Voo {
public String getOrigem()
public String getDestino()
public String getData()
public int getHorario()
public int getAssentos()
public int getPreco()
public void setOrigem(String origem)
public void setDestino(String destino)
public void setData(String data)
public void setHorario(int horario)
public void setAssentos(int assentos)
public void setPreco(int preco)
public Voo(String origem, String destino, String data,
          int horario, int assentos, int preco)
public void reservar()
public String toString()}
```

Main

Main

Main

```
public static void main(String[] args){
        Scanner ler = new Scanner(System.in);
        System.out.print("Escolha o fluxo desejado:\n[0] Sequêncial\n[1]
Paralelo\n");
        int fluxo = ler.nextInt();
        ler.close();

        CSV csv = new CSV();
        Funcoes funcoes = new Funcoes();

        List<Cliente> clientes =
csv.leitorClientes("Plataforma/src/main/csv/clientes_1000.csv");
        List<Hotel> hoteis = csv.leitorHoteis("Plataforma/src/main/csv/hoteis.csv");

        List<Voo> voos = csv.leitorVoos("Plataforma/src/main/csv/voos.csv");

        long startTime, endTime;
```

Main

```
        if (fluxo == 0){
            startTime = System.nanoTime();
            String resp = "Plataforma/src/testes/test.csv";
            for (Cliente c : clientes){
                int h = funcoes.melhorHotel(hoteis, c);
                List<Voo> v = funcoes.melhorCaminho(c.getSaida(), c.getChegada(),
voos);
                if (h != -1 && v.size() > 0)
                    csv.escreverFinal(c.reservar(hoteis.get(h), v), resp);
                else
                    csv.escreverFinal(c.toString(), resp);
            }
            endTime = System.nanoTime();
```

Main

```
        } else {
            startTime = System.nanoTime();
            for (Cliente c : clientes){
                Thread c1 = new Thread(new Paralelo(c, hoteis, voos, csv, funcoes));

                c1.start();
            }
            endTime = System.nanoTime();
        }

        long duracao = (endTime - startTime)/1000000;
        System.out.println("Tempo de execução: " + duracao + " milisegundos");
    }
```

Paralelo

Paralelo

Paralelo

```
public class Paralelo implements Runnable {
public Paralelo(Cliente cliente, List<Hotel> hoteis, List<Voo> voos,
                CSV csv, Funcoes funcoes)
public void run()}
```

Tempos de execução

	100 clientes	1000 clientes	10000 clientes
Seq:	*97 ms*	*463 ms*	*3.3 s*
Par:	*130 ms*	*437 ms*	*1.8 s*

Conclusão

Obrigado