

**UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS DE QUIXADÁ**



**Relatório  
Mototáxi.net**

Disciplina: Redes e Sistemas Distribuídos – 2018.2

Equipe: Guilherme Rodrigues

Victor Lucas

Rafael Lima

Pedro Augusto

Quixadá - CE

## **1. Visão geral do serviço do remoto**

Nosso serviço tem como objetivo principal oferecer uma forma de Clientes chamarem Mototaxistas que estejam em um local determinado para uma viagem até seu destino. O Cliente para utilizar esse serviço deverá fazer um cadastro no sistema e indicar o ponto de partida da viagem.

O Cliente poderá ver a lista de Mototaxistas disponíveis no local em que ele indicou e verá as informações dos Mototaxistas.

O Mototaxista para oferecer seu serviço de viagem terá que fazer um cadastro no sistema afim de indicar sua localização atual e mostrar suas informações para o Cliente.

## **2. Descrição dos Métodos Remotos**

### **2.1 Fazer cadastro Cliente**

Descrição: Esse método permite com que o cliente faça um cadastro no sistema.

Entrada: nome e telefone.

Saída: Verdadeiro ou Falso.

### **2.2 Fazer cadastro Mototaxista.**

Descrição: Esse método permite com que o Mototaxista faça um seu cadastro no sistema.

Entrada: Nome, telefone e praça

Saída: Verdadeiro ou Falso.

### **2.3 Editar cadastro Cliente**

Descrição: Esse método permite com que o Cliente altere seus dados no sistema.

Entrada: Seu ID, novo nome e novo telefone.

Saída: Verdadeiro ou Falso.

### **2.4 Editar cadastro Mototaxista**

Descrição: Esse método permite com que o Mototaxista altere seus dados no sistema.

Entrada: Seu ID, novo nome, novo telefone e nova praça.

Saída: Verdadeiro ou Falso.

### **2.5 Solicitar Corrida.**

Descrição: O Cliente solicita uma corrida

Entrada:

Saída: Lista com os Mototaxistas disponíveis

### **2.6 Ver Mototaxistas Disponíveis**

Descrição: O Cliente ver os Taxistas disponíveis no local indicado.

Entrada: Praça

Saída: Lista de Mototaxistas disponíveis.

## 2.7 Deletar conta Cliente

Descrição: O Cliente irá apagar sua conta.

Entrada: Seu ID

Saída: Verdadeiro ou Falso.

## 2.8 Deletar conta Mototaxista

Descrição: O Mototaxista irá apagar sua conta.

Entrada: Seu ID

Saída: Verdadeiro ou Falso.

## 2.9 Listar Clientes

Descrição: Essa funcionalidade permite ver todos os usuários cadastrados no sistema

Entrada:

Saída: Lista com todos os Clientes.

## 3. Descrição dos Dados transmitidos

--- MOTOTÁXI ---

\* Cadastro de mototáxi:

Envio:

```
{
  "method": 'Mototaxi.cadastrar',
  "data": {
    "id_mototaxi": "",
    "nome": "",
    "telefone": "",
    "praca": ""
  }
}
```

Resposta:

```
{
  "message": 'Mototáxi cadastrado com sucesso',
  "data": {}
}
```

\* Listar de mototáxi:

Envio:

```
{
  "method": 'Mototaxi.listar',
  "data": {}
}
```

Resposta:

```
{
  "message": "",
  "data": [
    0: {
      "id_mototaxi": ...,
      "nome": ...,
      "telefone": ...,
      "praca": ...,

    }
    ...
  ]
}
```

\* Editar mototáxi

Envio:

```
{
  "method": 'Mototaxi.editar',
  "data": {
    "id_mototaxi": "",
    "nome": "",
    "telefone": "",
    "praca": "",

  }
}
```

Resposta:

```
{
  "message": 'Mototáxi editado com sucesso',
  "data": {}
}
```

### \* Remover mototáxi

Envio:

```
{
  "method": 'remove.Mototaxi',
  "data": {
    "id_mototaxi": "",
  }
}
```

Resposta:

```
{
  "message": 'Mototáxi removido com sucesso',
  "data": {}
}
```

--- CLIENTE ---

### \* Cadastro de cliente:

Envio:

```
{
  "method": 'Cliente.cadastrar',
  "data": {
    "id_cliente": "",
    "nome": "",
    "telefone": ""
  }
}
```

Resposta:

```
{
  "message": 'Cliente cadastrado com sucesso',
  "data": {}
}
```

\* Listar de cliente:

Envio:

```
{
  "method": 'Cliente.listar',
  "data": {}
}
```

Resposta:

```
{
  "message": "",
  "data": [
    0: {
      "id_cliente": ...,
      "nome": ...,
      "telefone": ...
    },
    ...
  ]
}
```

\* Remover cliente:

Envio:

```
{
  "method": 'Cliente.remover',
  "data": {
    "id_cliente": "
  }
}
```

Resposta:

```
{
  "message": 'Cliente removido com sucesso',
  "data": {}
}
```

#### \* Editar cliente

Envio:

```
{
  "method": 'Cliente.editar',
  "data": {
    "id_cliente": "",
    "nome": "",
    "telefone": "",
  }
}
```

Resposta:

```
{
  "message": 'Cliente editado com sucesso',
  "data": {}
}
```

## 4. Descrição das classes implementadas nos lados Cliente e Servidor

### 4.1 Cliente

#### 4.1.1 Mototaxi

Proxy Mototaxis: É Responsável por se comunicar com o TCP Cliente, afim de simular para o Usuário (Mototaxista) que ele está acessando diretamente o serviço, quando na verdade ele está acessando remotamente.

TCP Cliente: Faz a comunicação com o TCP Server.

ControleParseJson: É responsável pela serialização e desserialização dos objetos.

Conexão: É responsável por estabelecer a conexão entre Cliente e Servidor

#### 4.1.2 Cliente

Proxy Cliente: É Responsável por se comunicar com o TCP Cliente, afim de simular para o Usuário (Cliente) que ele está acessando diretamente o serviço, quando na verdade ele está acessando remotamente

TCP Cliente: Faz a comunicação com o TCP Server.

ControleParseJson: É responsável pela serialização e desserialização dos objetos.

Conexão: É responsável por estabelecer a conexão entre Cliente e Servidor

#### **4.2 Servidor**

TCP Server: É responsável por responder às requisições do usuário.

Despachante: É responsável por chamar um método de acordo com a requisição.

Esqueleto Mototaxi:

Esqueleto Cliente:

Controlador Mototaxi: Essa classe executa de fato o serviço oferecido ao Mototaxista, ela é responsável por dar a resposta para o usuário, de acordo com sua requisição.

Controlador Cliente: Essa classe executa de fato o serviço oferecido ao Cliente, ela é responsável por dar a resposta para o usuário, de acordo com sua requisição.

Mototaxi: Contém os atributos do Mototaxista.

Cliente: Contém os atributos do Cliente.

ControleParseJson: É responsável pela serialização e desserialização dos objetos.

Conexão: É responsável por estabelecer a conexão entre Cliente e Servidor

#### **5. Descrever o modelo de falhas**

Fizemos tratamentos para os possíveis erros de Entrada e Saída verificando se os campos foram preenchido de forma correta, tendo em vista que um Cliente tem que ter todos os seus atributos preenchidos corretamente. As exceções de sistema como EOFException, IOException, UnknownHostException e outras, foram tratadas para dar uma experiência mais agradável ao usuário.