

# Aluguel de bicicletas compartilhadas

T3.6

## 1. LISTA DE OBJETOS, ATRIBUTOS E MÉTODOS

### ❖ Usuário

❖ **Atributos:** id, nome, sobrenome, email, endereço, telefone, número de CPF, plano de assinatura, quilômetros pedalados.

❖ **Métodos:**

- cadastraUsuario (String nome, String sobrenome, String email, Endereco endereco, Telefone telefone, int cpf, PlanoAssinatura plano, int kmPedalado): boolean
- nomeCompleto (): String
- email (): String
- telefone (): String
- cpf (): String
- plano (): String
- kmPedalado (): String
- addKmPedalado (float km): boolean
- editaUsuario (String nome, String sobrenome, String email, Endereco endereco, Telefone telefone, int cpf, PlanoAssinatura plano, int kmPedalado): boolean
- deleteUsuario (): boolean

### ❖ Telefone

❖ **Atributos:** ddd, número.

❖ **Métodos:**

- cadastraTelefone (int ddd, int numero): boolean
- ddd (): String
- numero (): String

- 
- editaTelefone (int ddd, int numero): boolean
  - deletaTelefone (): boolean

## ❖ Plano

- ❖ **Atributos:** tipo do plano, data de início
- ❖ **Métodos:**
  - tipo(): String
  - deletaPlano (): boolean

## ❖ PlanoPago

- ❖ **Atributos:** valor
- ❖ **Métodos:**
  - Tipo (): String
  - dataInicio (): String
  - editaPlano (String tipo, Date dataInicio, Date dataVencimento): boolean

## ❖ PlanoFree

## ❖ Bicicleta

- ❖ **Atributos:** id único, status, tipo, aro, cor, peso, total de quilômetros pedalados.
- ❖ **Métodos:**
  - cadastraBicicleta (String status, int aro, String cor, float peso, int kmPedalado): boolean + status(): String
  - tipo (): String
  - aro (): String
  - cor (): String
  - Peso (): String
  - kmPedalado (): String
  - addKmPedalado (float km): boolean
  - deletaBicicleta (): boolean

## ❖ Corrida

- ❖ **Atributos:** id, usuário, tempo de uso, data da utilização, bicicleta utilizada, quilômetros pedalados.
- ❖ **Métodos:**
  - cadastraCorrida (Datetime tempoUso, Date dataUso, Bicicleta bicicletaUtilizada, int kmPedalado): boolean
  - tempoUso (): String
  - dataUso (): String

- 
- bicicletaUtilizada (): String
  - kmPedalado (): String
  - addKmPedalado (float km): boolean
  - editaCorrida (Datetime tempoUso, Date dataUso, Bicicleta bicicletaUtilizada, int kmPedalado): boolean
  - deletaCorrida (): boolean

## ❖ Estação

❖ **Atributos:** id, nome, endereço, número de vagas, vagas disponíveis.

❖ **Métodos:**

- cadastraEstacao (String nome, Endereco endereco, int numVagas, int numVagasDisponiveis): boolean
- nome (): String
- endereço (): String
- numVagas (): String
- numVagasDisponiveis (): String
- editaEstacao (String nome, Endereco endereco, int numVagas, int numVagasDisponiveis): boolean
- editaEstacao (int numVagasDisponiveis): boolean
- deletaEstacao (): boolean

## ❖ Endereço

❖ **Atributos:** cep, país, estado, bairro, rua/quadra, número.

❖ **Métodos:**

- cadastraEndereco (int cep, String pais, String estado, String bairro, String ruaQuadra, int numero): boolean
- cep (): String
- pais (): String
- estado (): String
- bairro (): String
- ruaQuadra (): String
- numero (): String
- editaEndereco (int cep, String pais, String estado, String bairro, String ruaQuadra, int numero): boolean
- deletaEndereco (): boolean

---

## 2. LISTA DE FUNCIONALIDADES DO SOFTWARE

### 2.1 Requisitos Funcionais

- ❖ **RF1:** Deve ser possível cadastrar, visualizar, editar e deletar (realizar CRUD) de Usuário.
- ❖ **RF2:** Deve ser possível realizar CRUD de Telefone.
- ❖ **RF3:** Deve ser possível realizar CRUD de Plano de Assinatura.
- ❖ **RF4:** Deve ser possível realizar CRUD de Endereço.
- ❖ **RF5:** Deve ser possível realizar CRUD da Corrida.
- ❖ **RF6:** Deve ser possível buscar a corrida através do dia.
- ❖ **RF7:** Deve ser possível buscar pelo nome, endereço ou lugares vagos da Estação.
- ❖ **RF8:** Deve ser possível listar as corridas feitas pelo usuário.
- ❖ **RF9:** Deve ser possível listar os usuários com mais quilômetros pedalados.
- ❖ **RF10:** O software deve possuir um conjunto de dados pré-cadastrados

### 2.2 Requisitos Não Funcionais

- ❖ **RNF1:** O software deve ser desenvolvido em Java;
- ❖ **RNF2:** O software deve ser desenvolvido utilizando o paradigma orientado a objetos;
- ❖ **RNF3:** A interação com o usuário deverá ser feita por meio de interface gráfica;
- ❖ **RNF4:** O software desenvolvido será para ambiente desktop;
- ❖ **RNF5:** Os valores dos planos de assinatura devem estar em reais (R\$);
- ❖ **RNF6:** Os valores de distância percorrida devem estar em quilômetros (Km);
- ❖ **RNF7:** Os elementos gráficos do aplicativo devem seguir uma identidade visual.

### 2.3 Prioridade dos Requisitos

Prioridade	Requisito(s)
1	RF1, RF2, RF3, RF4, RF5, RF10, RNF1, RNF2, RNF3, RNF4, RNF5, RNF6, RNF7.
2	RF6, RF7
3	RF8, RF9