

Aluguel de bicicletas compartilhadas

T3.6

1. LISTA DE OBJETOS, ATRIBUTOS E MÉTODOS

❖ **Usuário**

- ❖ **Atributos:** nome, sobrenome, email, endereço, telefone, cpf, plano, quilômetros pedalados.
- ❖ **Métodos:**
 - Usuario (String nome, String sobrenome, String email, Endereco endereco, Telefone telefone, String cpf, Plano plano, int kmPedalado): construtor
 - getNomeCompleto (): String
 - getEmail (): String
 - getEndereco (): Endereco
 - getTelefone (): Telefone
 - getCpf (): String
 - getPlano (): String
 - getKmPedalado (): int
 - setNome (String): void
 - setSobrenome (String): void
 - setEmail (String): void
 - setEndereco (Endereco): void
 - setTelefone (Telefone): void
 - setCpf (String): void
 - setPlano (Plano): void
 - setKmPedalado (): void
 - addKmPedalado (int km): void
 - editaUsuario (String nome, String sobrenome, String email, Endereco endereco, Telefone telefone, String cpf, int kmPedalado): void

-
- `deletaUsuario ()`: void

❖ **Telefone**

❖ **Atributos:** ddd, número.

❖ **Métodos:**

- `Telefone (String ddd, String numero)`: construtor
- `getDDD ()`: String
- `getNumero ()`: String
- `setDDD (String)`: void
- `setNumero (String)`: void
- `editaTelefone (String ddd, String numero)`: void
- `deletaTelefone ()`: void

❖ **Plano (abstract)**

❖ **Atributos:** tipo, data de início.

❖ **Métodos:**

- `Plano ()`: construtor
- `getTipo ()`: String
- `getDataInicio ()`: Date
- `setTipo (String tipo)`: void
- `setDataInicio (Date dataInicio)`: void
- `editaPlano (String tipo, Date dataInicio)`: void
- `deletaPlano ()`: void

❖ **PlanoPago (extends Plano)**

❖ **Atributos:** valor

❖ **Métodos:**

- `PlanoPago (String tipo, Date dataInicio, double valor)`: construtor
- `getValor ()`: double
- `setValor (double valor)`: void
- `editaPlano (String tipo, Date dataInicio, Date dataVencimento)`: void

❖ **PlanoAnual (extends PlanoPago)**

❖ **Atributos:**

❖ **Métodos:**

- `PlanoAnual (Date dataInicio)`: construtor

❖ **PlanoMensal** (*extends PlanoPago*)

❖ **Atributos:**

❖ **Métodos:**

- PlanoMensal (Date dataInicio): construtor

❖ **PlanoFree** (*extends Plano*)

❖ **Atributos:**

❖ **Métodos:**

- PlanoFree (Date dataInicio): construtor

❖ **Bicicleta**

❖ **Atributos:** status, tipo, aro, cor, peso, total de quilômetros pedalados.

❖ **Métodos:**

- Bicicleta (boolean status, int aro, String cor, float peso, int kmPedalado): construtor
- getStatus(): boolean
- getTipo (): String
- getAro (): String
- getCor (): String
- getPeso (): String
- getKmPedalado (): int
- setStatus (boolean status): void
- setTipo (String tipo): void
- setAro (int aro): void
- setCor (String cor): void
- setPeso (float peso): void
- setKmPedalado (int kmPedalado): void
- addKmPedalado (int km): void
- deletaBicicleta (): void

❖ **Corrida**

❖ **Atributos:** usuário, tempo de uso, data da utilização, bicicleta utilizada, quilômetros pedalados.

❖ **Métodos:**

- Corrida (Usuario usuario, long tempoUso, Date dataUso, Bicicleta bicicletaUtilizada, int kmPedalado): construtor
- getUsuario (): String
- getTempoUso (): String
- getDataUso (): Date

-
- getBicicletaUtilizada (): String
 - getKmPedalado (): int
 - setUsuario (Usuario usuario): void
 - setTempoUso (long tempoUso): void
 - setDataUso (Date data): void
 - setBicicletaUtilizada (Bicicleta bike): void
 - setKmPedalado (int km): void
 - addKmPedalado (int km): void
 - editaCorrida (Usuario usuario, Datetime tempoUso, Date dataUso, Bicicleta bicicletaUtilizada, int kmPedalado): void
 - deleteCorrida (): void

❖ Estação

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

❖ **Métodos:**

- Estacao (String nome, Endereco endereco, int numVagas, int numVagasDisponiveis): construtor
- getNome (): String
- getEndereço (): Endereco
- getNumVagas (): int
- getNumVagasDisponiveis (): int
- setNome (String nome): void
- setEndereço (Endereco end): void
- setNumVagas (int i): void
- editaEstacao (String nome, Endereco endereco, int numVagas, int numVagasDisponiveis): void
- editaEstacao (int numVagasDisponiveis): void
- deleteEstacao (): void

❖ Endereço

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

❖ **Métodos:**

- Endereco (String cep, String pais, String estado, String bairro, String ruaQuadra, String numero): construtor
- completo (): String
- getCep (): String
- getPais (): String
- getEstado (): String
- getBairro (): String

-
- getRuaQuadra (): String
 - getNumero (): String
 - setCep (String cep): void
 - setPais (String pais): void
 - setEstado (String Estado): void
 - setBairro (String bairro): void
 - setRuaQuadra (String ruaQuadra): void
 - setNumero (String num): void
 - editaEndereco (String cep, String pais, String estado, String bairro, String ruaQuadra, String numero): void
 - deletaEndereco (): void

❖ Dados

❖ **Atributos:** Usuario[], qtdUsuario, Endereco[], qtdEndereco, Telefone[], qtdTelefone, Bicicleta[], qtdBicicleta, Estacao[], qtdEstacao, Corrida[], qtdCorrida.

❖ **Métodos:**

- carregaDados (): void
- getQtdUsuarios (): Usuario[]
- addUsuario (Usuario u, int pos): void
- addEndereco (Endereco end, int pos): void
- getEndereco (int pos): Endereco
- addTelefone (Telefone tel, int pos): void
- getTelefone (int pos): Telefone
- getQtdBicicleta (): int
- getBicicleta (): Bicicleta[]
- addBicicleta(Bicicleta b, int pos): void
- getQtdEstacoes (): int
- getEstacoes (): Estacao[]
- getCorrida (int pos): String

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 listar as informações das 10 últimas corridas realizadas no app.
 - ❖ **RF7:** Deve ser possível buscar todas as informações da estação pelo nome da estação.
 - ❖ **RF8:** Deve ser possível buscar todas as informações da estação pelo número de vagas disponíveis da estação.
 - ❖ **RF9:** 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);

2.3 Prioridade dos Requisitos

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