



Departamento de Ciência da computação – Universidade de Brasília (UnB)  
**Disciplina: Técnicas de Programação 1 – Turma: B Professora: Roberta  
Barbosa Oliveira Grupo 17: Isaac Weber Borges Guedes – 200019821**

## **FAST VEHICLES “O jeito rápido e fácil de alugar o seu veículo”**

### **1. Descrição do Problema:**

A FastVehicle é uma loja virtual onde se pode alugar o veículo desejado por meio da realização de um cadastro em seu aplicativo. Com ela, é possível alugar seu veículo de forma rápida, com cauções acessíveis e sem muita burocracia. É necessário apenas que sua Carteira Nacional de Habilitação (CNH) esteja regular e que possua algum cartão de crédito para pagar o caução. Preenchendo tais requisitos, você estará apto a ter um excelente veículo, num valor que cabe no seu bolso, para o seu uso como for desejado. Há possíveis veículos de vários tipos: carros sedãs, picapes, SUVs, esportivos, conversíveis; motos street, custom, trail, Sport e muito mais tipos de carros e motos.

O aplicativo é o coração da FastVehicle, pois, a partir dele, é que os clientes podem alugar o carro desejado apenas indo até um dos galpões registrados no país para pegá-lo.

Os locadores que desejarem criar a sua locadora podem acessar o aplicativo, mas devem seguir as regras de negócio da Fast Vehicles.

Para o funcionamento da Fast Vehicles, é preciso que existam locadores com veículos acessíveis e que sigam as regras de negócio instauradas, que são muito simples.

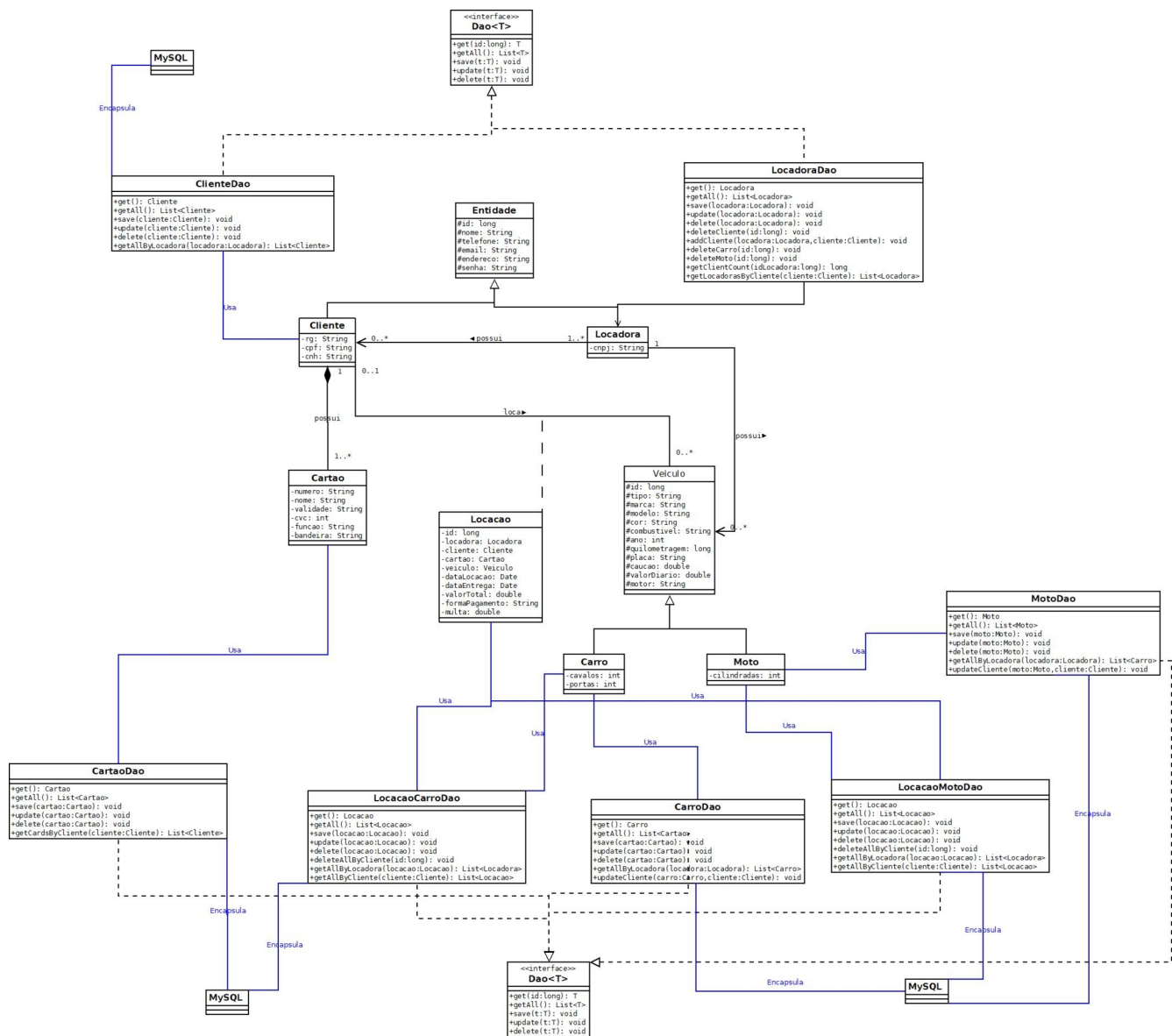
### **2. Regras de Negócio:**

Serão consideradas as seguintes regras de negócio:

- O cliente precisa ter obrigatoriamente a sua Carteira Nacional de Habilitação em dia.
- O cliente só pode fazer a reserva de um veículo caso tenha um cartão de crédito para ser feito o caução.

- O cliente precisa informar a quantidade de pessoas que estarão no veículo e quem será o motorista.
- A locadora precisa informar com precisão quais são os veículos disponíveis de acordo com a solicitação do cliente.
- O cliente precisa informar obrigatoriamente qual será o destino.
- Caso o pagamento seja feito à vista, o cliente terá um desconto de 10%.
- Não se pode cobrar um caução ou preço diário desonesto pelo veículo.

### 3. Diagrama de Classes



### 3.1. Descrição das Classes & Interfaces

#### Classe Abstrata Entidade:

Classe que representa as duas entidades na ação – Cliente e Locadora. Ela guarda vários atributos pertencentes aos dois: id, nome, telefone...

#### Classe Cliente:

Classe que representa o cliente. Ela estende a classe Entidade e coloca alguns atributos a mais – rg, cpf e cnh.

#### Classe Locadora:

Representa a locadora envolvida na ação. Estende a classe Entidade e coloca o atributo cnpj.

#### Classe Cartao

Representa o cartão do cliente. Possui vários atributos pertinentes a um cartão comum – número, cvc, função – débito ou crédito, etc.

#### Classe Abstrata Veiculo:

Classe de base para os veículos, carro e moto. Ela possui os atributos que um carro comum e uma moto possuem.

#### Classe Carro:

Classe que representa o carro. Estende a classe Veiculo e possui os atributos: cavalos e portas.

#### Classe Moto:

Classe que representa a moto. Estende a classe Veiculo e tem o atributo: cilindradas.

#### Classe Locacao:

Classe que representa a locação. Ela possui todos os atributos necessários para se registrar uma locação no sistema e ficar registrada no banco de dados.

#### Interface Genérica Dao<T>:

Interface base para a manipulação do banco de dados, representa o acesso dos objetos por meio de um determinado modelo, T, vinda das palavras: Data Access Object (DAO). “Todas as classes que implementam essa interface servem para encapsular os dados, não deixando assim código SQL misturado com código Java.”

Possui os métodos de salvar, atualizar, deletar, pegar e pegar todos, que são, respectivamente, save, update, delete, get e getAll.

### Classe ClienteDao:

Classe que implementa a interface `Dao<T>`, e admite como T, o valor do modelo representado pela classe `Cliente`. Essa classe serve para manipular as requisições do banco de dados que envolvem a classe `Cliente`.

Além dos métodos implementados pela interface `Dao<T>`, essa classe possui o método *`getAllByLocadora`*, que retorna uma lista de clientes a partir de um objeto da classe `Locadora`.

### Classe LocadoraDao:

Classe que implementa a interface `Dao<T>`, e admite como T, o valor do modelo representado pela classe `Locadora`. Essa classe serve para manipular as requisições do banco de dados que envolvem a classe `Locadora`.

- Métodos independentes da implementação de `Dao<T>`:
  1. *`addCliente`*: Adiciona cliente no banco de dados, mas em uma tabela diferente da tabela de clientes. Identificando a união da locadora com o cliente.
  2. *`deleteCliente`*: Deleta cliente no banco de dados, mas em uma tabela diferente da tabela de clientes. Identificando o fim da união da locadora com o cliente.
  3. *`deleteCarro`*: Deleta carro do banco de dados e de suas locações. Indicando o fim do carro e de suas locações.
  4. *`deleteMoto`*: Deleta moto da mesma forma que deletou o carro.
  5. *`getClientCount`*: Pega a quantidade de clientes desta locadora.
  6. *`getLocadorasByCliente`*: Pega todas as locadoras a partir de um cliente.

### Classe CartaoDao:

Classe que implementa a interface `Dao<T>`, e admite como T, o valor do modelo representado pela classe `Cartao`. Essa classe serve para manipular as requisições do banco de dados que envolvem a classe `Cartao`.

Além dos métodos implementados pela interface `Dao<T>`, implementa um método chamado *`getCardsByCliente`*, que retorna todos os cartões a partir de um dado cliente.

### Classe MotoDao:

Classe que implementa a interface `Dao<T>`, e admite como T, o valor do modelo representado pela classe Moto. Essa classe serve para manipular as requisições do banco de dados que envolvem a classe Moto.

Além dos métodos implementados pela interface `Dao<T>`, essa classe possui os métodos *getAllByLocadora* e *updateCliente*, que pega todas as motos por locadora e atualiza o cliente da moto, respectivamente.

#### Classe CarroDao:

Classe que implementa a interface `Dao<T>`, e admite como T, o valor do modelo representado pela classe Carro. Essa classe serve para manipular as requisições do banco de dados que envolvem a classe Carro.

Além dos métodos implementados pela interface `Dao<T>`, possui os mesmos métodos da moto, mas desempenhando o papel com a classe Carro.

#### Classe LocacaoCarroDao:

Classe que implementa a interface `Dao<T>`, e admite como T, o valor do modelo representado pela classe Locacao e admitindo o veículo como sendo um objeto da classe Carro. Essa classe serve para manipular as requisições do banco de dados que envolvem a classe Locacao juntamente com a classe Carro. Simplificado, envolve a locação de um carro.

Além dos métodos implementados pela interface `Dao<T>`, possui os métodos *deleteAllByCliente*, deleta todos por cliente, *getAllByLocadora*, pega todos por locadora, *getAllByCliente*, pega todos por cliente.

#### Classe LocacaoMotoDao:

Classe que implementa a interface `Dao<T>`, e admite como T, o valor do modelo representado pela classe Locacao e admitindo o veículo como sendo um objeto da classe Moto. Essa classe serve para manipular as requisições do banco de dados que envolvem a classe Locacao juntamente com a classe Moto. Simplificado, envolve a locação de uma moto.

Além dos métodos implementados pela interface `Dao<T>`, possui os métodos *deleteAllByCliente*, deleta todos por cliente, *getAllByLocadora*, pega todos por locadora, *getAllByCliente*, pega todos por cliente.

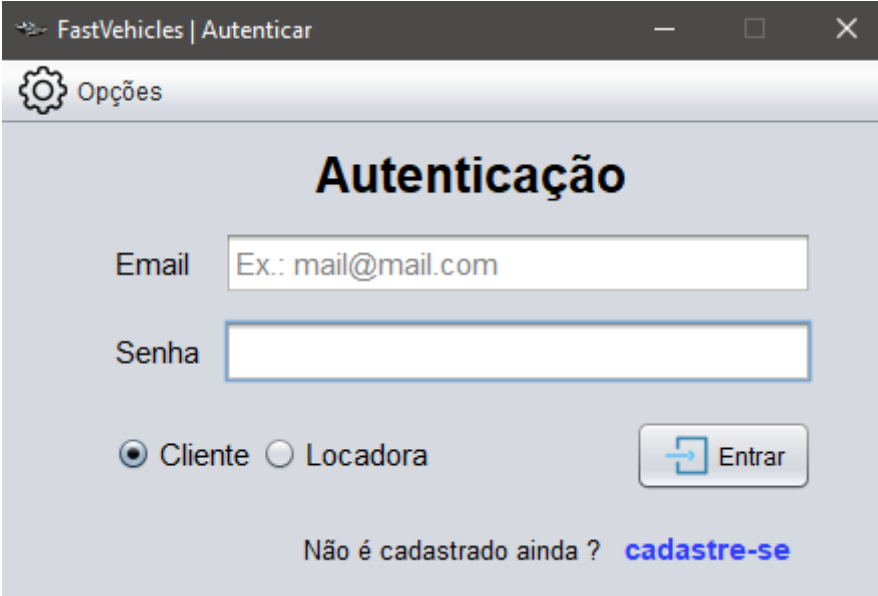
#### Classe representativa MySQL:

Essa classe é apenas representativa, que denota a existência do banco de dados e seu tipo para facilitar o entendimento do diagrama. Essa classe representa o banco de dados, com gerenciador MySQL.

#### 4. Telas

- São muitas telas, bastante intuitivas.
- Pelo fato de serem muitas, não serão exibidas todas.

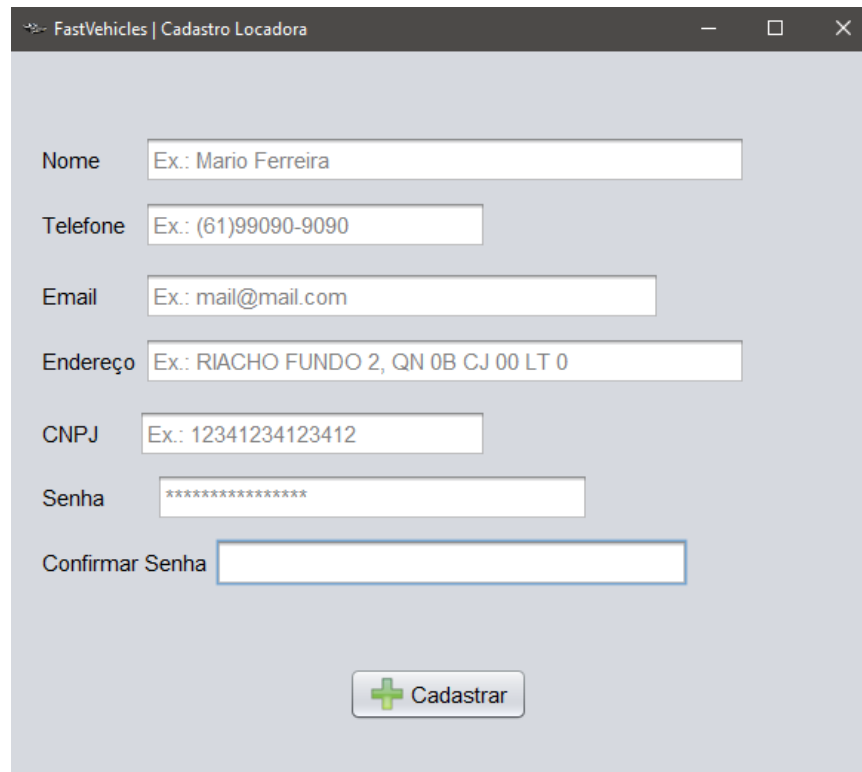
##### 4.1. Tela de autenticação



The screenshot shows a web browser window titled "FastVehicles | Autenticar". Below the title bar is a header with a gear icon and the word "Opções". The main content area has a light gray background and is titled "Autenticação" in bold black text. It contains two input fields: "Email" with a placeholder "Ex.: mail@mail.com" and "Senha" (Password). Below these fields are two radio buttons labeled "Cliente" (selected) and "Locadora". To the right of the radio buttons is a button labeled "Entrar" with a blue arrow icon. At the bottom, there is a link that says "Não é cadastrado ainda ? cadastre-se".

Tela de autenticação para usuário sendo cliente ou locadora, basta selecionar o botão de rádio para o respectivo.

#### 4.2. Tela de Cadastro de Locadora



The image shows a web browser window titled "FastVehicles | Cadastro Locadora". The form contains the following fields and labels:

- Nome: Ex.: Mario Ferreira
- Telefone: Ex.: (61)99090-9090
- Email: Ex.: mail@mail.com
- Endereço: Ex.: RIACHO FUNDO 2, QN 0B CJ 00 LT 0
- CNPJ: Ex.: 12341234123412
- Senha: \*\*\*\*\*
- Confirmar Senha: (empty field)

At the bottom of the form is a button labeled "Cadastrar" with a green plus icon.

Preencha os dados e cria a sua locadora nova. Os campos possuem explicações para facilitar o cadastro.

### 4.3. Cadastro Cliente

The image shows a web application window titled "FastVehicles | Cadastro Cliente". The window contains a registration form with the following fields and labels:

- Nome: Ex.: Mario Ferreira
- Telefone: Ex.: (61)99090-9090
- Email: Ex.: mail@mail.com
- Endereço: Ex.: RIACHO FUNDO 2, QN 0B CJ 00 LT 0
- RG: Ex.: 000000000
- CPF: Ex.: 000000000-00
- CNH: Ex.: 2222222222
- Senha: \*\*\*\*\*
- Confirmar Senha: (empty field)

At the bottom of the form is a button labeled "Cadastrar" with a green plus icon.

Cadastro de cliente. Preencha os dados e seja um novo cliente. Basta acessar a conta após o cadastro.




#### 4.4. Entrada do Cliente

FastVehicles | Entrada Cliente - NICOLAS KALEB BORGES GUEDES


Gerenciar Opções

### Escolha uma locadora

 Pesquise cliente pelo campo desejado

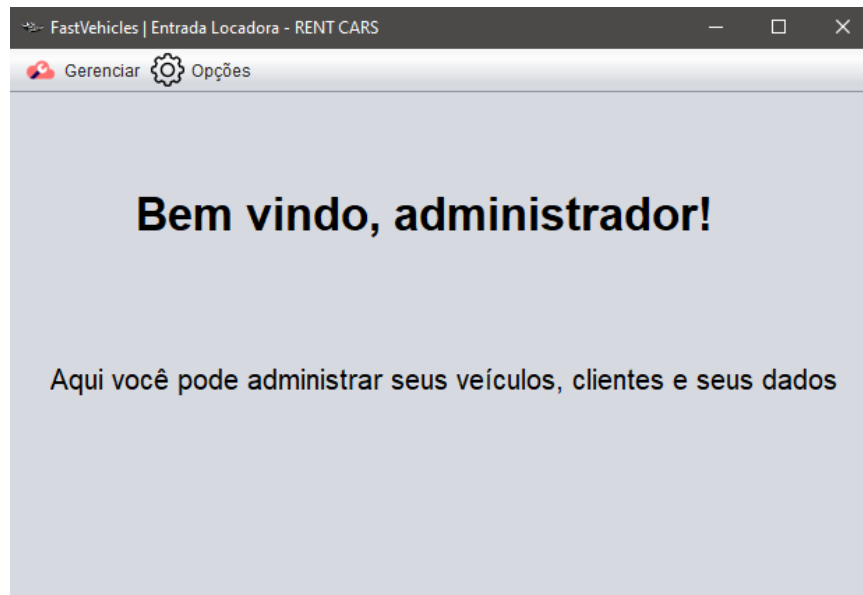
☒ id ☐ cnpj ☐ telefone ☐ email ☐ nome

Id	Nome	Telefone	Email	Endereco	cnpj
6	RENT CARS	6133333333	rentcars@gmail.com	QNL 06, SETOR DE...	718324322322212
7	fast cars	6133333333	fastcars@gmail.com	setor comercial sul, ...	12345678912354

 Acessar

Após o cliente ter cadastro, basta entrar. Esse é um exemplo de um cliente que se cadastrou e assim pode acessar alguma das locadoras registradas.

#### 4.5. Entrada da Locadora



Após criar uma locadora, é possível já acessar seu painel e adicionar seus veículos e manipular seus clientes.

#### 4.6. Tela de Locação do Cliente

FastVehicles | Locação - gol g4

Modelo Veículo

Data Locação

Data Entrega


Forma de Pagamento

☒ Cartão

☐ Dinheiro

Selecione um cartão (Obrigatório)

Id	Numero	Banco	Validade	Cvc	Bandeira
9	123412341234124134	BB	02/27	266	VISA

 Confirmar Locação

Após o cliente cadastrar um cartão, em outros frames não exibidos, é possível locar algum veículo. Segue um exemplo de um cliente com um cartão já cadastrado que desejou locar um carro, Gol G4.

Acesse o Aplicativo para ver todas as telas.