



## Laboratório 2

### Seguradora - Cadastro de Clientes e Geração de Sinistros

MC322 - Programação Orientada a Objetos

Professores: Esther Colombini / Marcos Raimundo

PEDs: Lucas Castro / Rebeca Padovani / Athyrson Ribeiro

## 1. Descrição Geral

Em virtude do grande número de acidentes de trânsito no Brasil, há uma demanda por seguradoras de veículos automotores. Contudo, para tais seguradoras, há uma complexidade de modelagem de sistemas para análise de preços dos seguros de acordo com o perfil de seus clientes. Informações como idade, estado civil, localidade, perfil de trabalho impactam diretamente na formulação de preço. Outro ponto a ser mencionado é o histórico de sinistros (acidentes envolvidos) que um condutor pode possuir. Com base nessas informações, o tópico a ser trabalhado para este semestre na disciplina de Programação Orientada a Objetos (MC322) é a modelagem de um sistema de seguros. Ao longo dos laboratórios, seu sistema deverá ser incrementado para comportar novas funcionalidades à medida que o seu conhecimento for ampliado.

Nesse primeiro laboratório, você deve implementar o sistema de cadastro de clientes e de sinistros. Para o cadastro de clientes, é necessário também o cadastro de veículos dos clientes. Sendo assim, inicialmente o sistema deve compreender quatro (4) classes: **Cliente**, **Veículo**, **Seguradora** e **Sinistro**. A Figura 1 apresenta um diagrama de classes das classes a serem implementadas.

Um diagrama de classes é um tipo de diagrama UML (*Unified Modeling Language*). Esse diagrama representa a estrutura de um sistema orientado a objetos, demonstrando assim classes, interfaces, propriedades, métodos e suas associações dentro do sistema. O objetivo principal do diagrama de classes é fornecer uma visão geral da arquitetura do sistema, mostrando como as diferentes partes do sistema se relacionam entre si.

## 2. Objetivos

Os principais objetivos deste laboratório consistem em:

- Reconhecimento e familiaridade com o ambiente de desenvolvimento (IDE) e a linguagem a ser utilizada nesta disciplina (Java);
- Desenvolvimento das classes e seus respectivos atributos e métodos;
- Desenvolvimento da visibilidade dos atributos e métodos.

## 3. Atividades

As seguintes atividades são esperadas nesse laboratório:

- Criação de um projeto na IDE a sua escolha: Dê preferência para o Eclipse como IDE, contudo, outras IDEs (VSCode, Sublime, Atom, etc) podem ser utilizadas;
- Elaboração das classes **Cliente**, **Veículo**, **Seguradora** e **Sinistro**: seu projeto deve compreender estas quatro classes além de uma classe **Main** para a execução do projeto;
- Função de geração de IDs (identificadores) para a classe **Sinistro**: cada objeto da classe **Sinistro** deve conter um identificador único (id). Desenvolva uma função na classe **Sinistro** que gere um valor para esse identificador. Dica: você pode utilizar uma função randômica para esse fim;
- Definição dos métodos, atributos, bem como suas respectivas visibilidades: você deve definir qual visibilidade (public, private, protected) os atributos e métodos possuem;
- Métodos de acesso (*getters e setters*): defina os métodos de acesso aos atributos das classes.

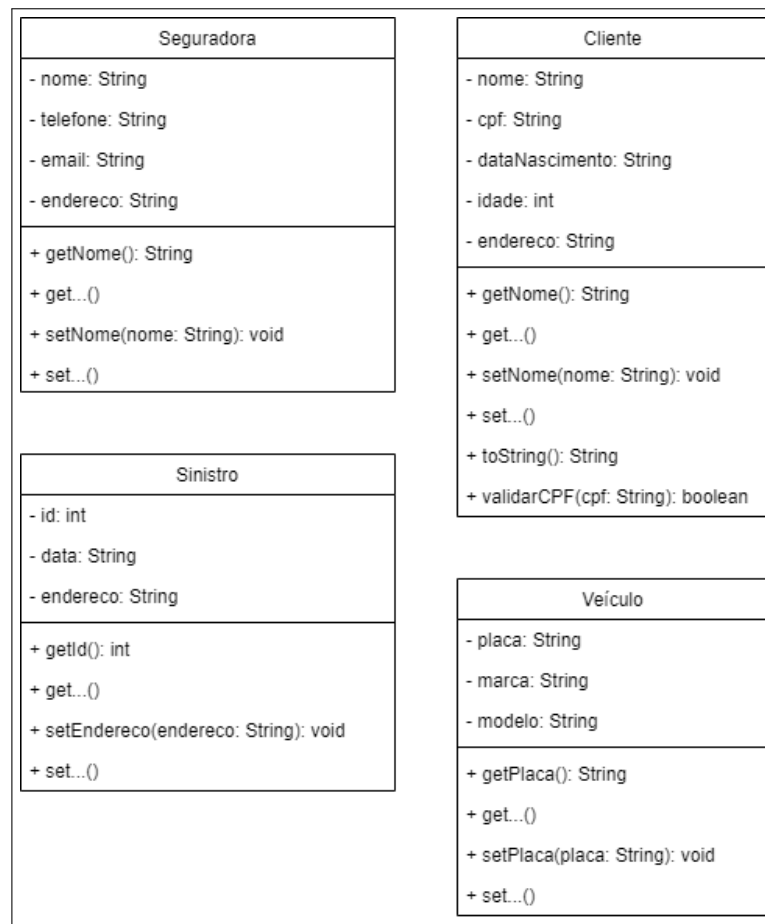


Figura 1: Diagrama de Classe

- Método *validarCPF()*: defina o método de validar o CPF de um cliente. Para isso você pode seguir os seguintes passos:
  1. Remova todos os caracteres não numéricos do CPF usando o método **replaceAll**;
  2. Verifique se o CPF tem 11 dígitos. Se não tiver, retorne false;
  3. Verifique se todos os dígitos são iguais. Se forem, retorne false;
  4. Calcule os dígitos verificadores usando o algoritmo apropriado;
  5. Verifique se os dígitos verificadores calculados são iguais aos dígitos verificadores do CPF. Se forem, retorne true. Caso contrário, retorne false.

## 4. Descrição das Classes

### Classe - Veículo

Classe responsável por armazenar as informações dos veículos segurados dos clientes. A classe é composta pelos seguintes atributos: **placa**, **marca** e **modelo**. Todos os atributos são do tipo String.

### Classe - Cliente

Essa classe compreende as informações do cliente segurado. Ela é composta de cinco atributos (**nome**, **cpf**, **dataNascimento**, **idade** e **endereco**). O atributo **idade** é do tipo inteiro (int), ao passo que os demais são todos do tipo String.

### Classe - Seguradora

A classe **Seguradora** tem como objetivo ter as informações da seguradora. Os atributos dessa classe são: **nome**, **telefone**, **email** e **endereco**. Todos os atributos são do tipo String.

```

1 public class Seguradora {
2     private String nome;
3     private String telefone;
4     private String email;
5     private String endereco;
6
7     // Construtor
8     public Seguradora(String nome, String telefone, String email, String endereco) {
9         this.nome = nome;
10        this.telefone = telefone;
11        this.email = email;
12        this.endereco = endereco;
13    }
14
15    // Getters e setters
16    public String getNome() {
17        return nome;
18    }
19
20    public void setNome(String nome) {
21        this.nome = nome;
22    }
23
24    public String getTelefone() {
25        return telefone;
26    }
27
28    public void setTelefone(String telefone) {
29        this.telefone = telefone;
30    }
31
32    public String getEmail() {
33        return email;
34    }
35
36    public void setEmail(String email) {
37        this.email = email;
38    }
39
40    public String getEndereco() {
41        return endereco;
42    }
43
44    public void setEndereco(String endereco) {
45        this.endereco = endereco;
46    }
47
48 }

```

Listing 1: Seguradora.java

## Classe - Sinistro

A Classe Sinistro corresponde em armazenar as informações relacionadas um evento em que o seguro foi acionado (acidente, furto, etc). Para isso, a Classe Sinistro possui três atributos (id, data e endereço). Data e endereço são do tipo String. Ao passo que o atributo id é um int que deverá ser gerado por uma função geradora de identificadores únicos.

## 5. Avaliação

Além da correta execução do laboratório, os seguintes critérios serão utilizados para a composição da nota do laboratório:

- Entrega realizada dentro do prazo estipulado;
- Qualidade do código desenvolvido (tabulação, comentários);
- Instanciação dos objetos na classe Main;
- Desenvolvimento correto dos métodos de acesso.

## 6. Entrega

- **A entrega do Laboratório é realizada exclusivamente via Github**<sup>1</sup> Para a submissão no Github, gere um release (tag) com a identificação do laboratório.
- Utilize os horários de laboratório e atendimentos para tirar eventuais dúvidas de submissão e também relacionadas ao desenvolvimento do laboratório.
- **Prazo de Entrega:** 28/03 - 14h

---

<sup>1</sup>O Laboratório 1 consistia no envio do seu repositório a ser utilizado na disciplina. Caso não tenha feito, procure os PEDs para informar seu repositório. Pois do contrário, seu laboratório não poderá ser avaliado.