

## Laboratório 6

### Seguradora - Interfaces, Tratamentos de Exceções e Arquivos

MC322 - Programação Orientada a Objetos

## 1 Descrição Geral

Nas atividades deste laboratório, iremos focar em conceitos de Orientação a Objetos vistos em classe, como: Interfaces, Leitura e Escrita de Arquivos e Tratamento de Exceções. A implementação desses conceitos serão os requisitos finais para o Sistema de Seguradora. Para ilustrar as novas implementações que estão incluídas neste laboratório, a Figura 1 apresenta o diagrama de classes<sup>1</sup>.

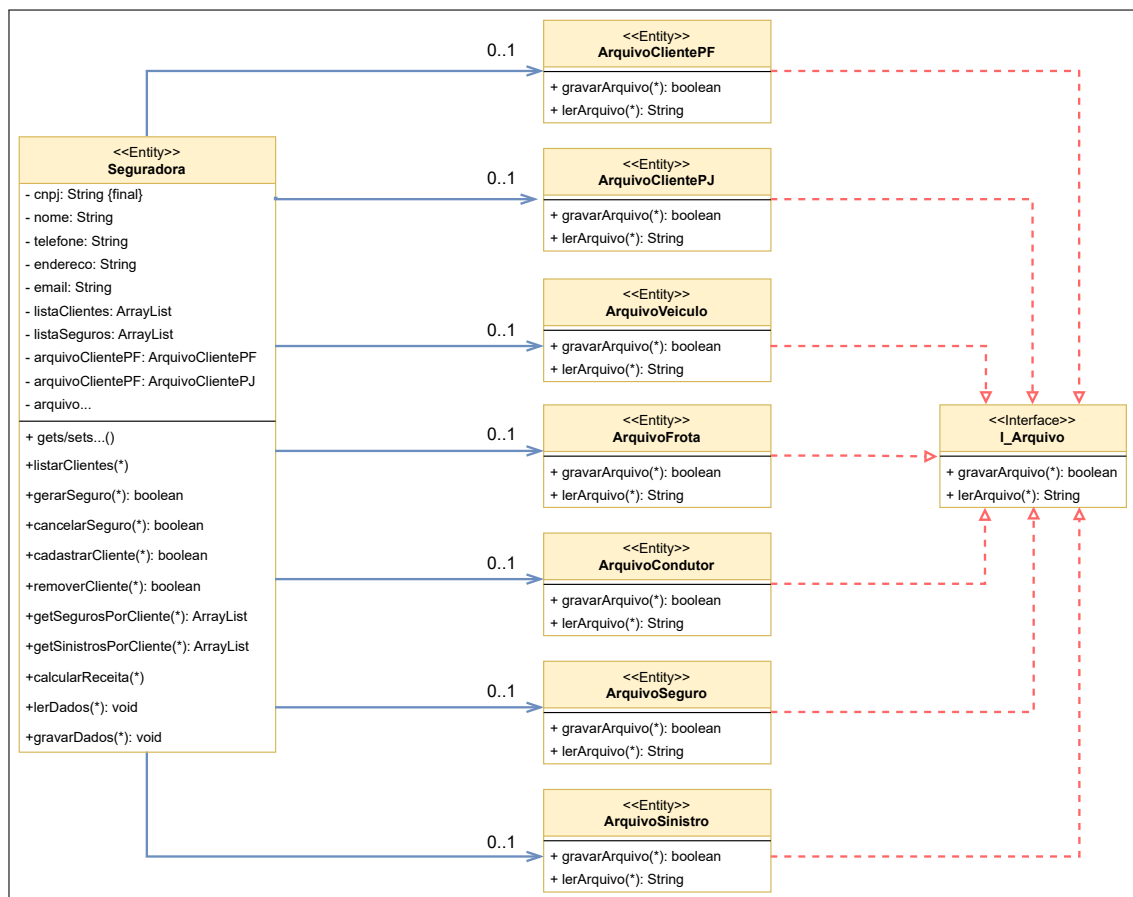


Figura 1: Diagrama de Classe - Sistema da Seguradora - I/O

Note que neste Laboratório (06), utilizaremos as classes do Laboratório anterior (05) e uma interface que deve ser implementada. Esse laboratório visa fixar os conceitos vistos até então e apresentar novos conceitos como:

1. **Tratamento de Exceções:** tratar exceções permite que situações de erro sejam estruturadas e controladas, evitando que o programa seja interrompido abruptamente. Com a leitura/escrita de arquivos, tratar exceções será importante para manter a robustez do programa, fornecer mensagens de erro úteis,

<sup>1</sup>Note que algumas das funcionalidades requeridas não estão explicitamente demonstradas no diagrama de classes. Tal abordagem visa um dos objetivos desse laboratório, o qual é a ampliação da capacidade de abstração por parte dos alunos em como aplicar os conceitos visto em aula para a resolução de problemas.

realizar ações de recuperação, fechar recursos adequadamente e comunicar erros para outras partes do sistema.

2. **Arquivos:** ao utilizar arquivos, é possível armazenar as informações de forma persistente, ou seja, mesmo quando o programa é encerrado, os dados são mantidos no arquivo. Isso permite que os dados sejam recuperados posteriormente (leitura), mantendo a integridade e a continuidade das informações. las estabelecem um conjunto de métodos que as classes devem implementar, definindo assim um comportamento esperado.
3. **Interfaces:** as interfaces estabelecem um conjunto de métodos que as classes devem implementar, de modo que cada classe tenha sua lógica específica para lidar com esse conjunto de métodos. Nesse contexto, os métodos realizarão a leitura e escrita de dados (por exemplo, a leitura/escrita de dados de clientes será diferente da leitura/escrita de dados dos veículos).

## 2 Objetivos

Os objetivos principais do Laboratório 6 são os seguintes:

- Consolidação dos conteúdos vistos nos labs anteriores;
- Capacidade prevenir erros do programa através do uso de Exceptions;
- Capacidade de interagir com arquivos;
- Implementação de interfaces para auxiliar no processo de leitura de arquivos para instanciação de objetos;

## 3 Atividades

As atividades a serem desenvolvidas para este Laboratório são as seguintes:

- Implementação de interface para I/O de arquivos;
- Leitura de dados dos arquivos:
  - Dados de Clientes (PF e PJ);
  - Dados de Veículos;
  - Dados de Frotas;
  - Dados de Condutores.
- Gravação de dados em arquivos:
  - Dados de Seguros;
  - Dados de Sinistros;
- Tratar exceções de leitura/escrita de arquivos, ou seja, garantir que a execução do programa não encerre por erros de leitura/escrita.

Na classe **AppMain**:

- Instanciar pelo menos 1 objeto de cada classe na ordem mais conveniente (é fortemente recomendado que se instancie múltiplos objetos para cada classe) para realizar a gravação de dados;
  - Seguros devem ser instanciados a partir do método apropriado da classe Seguradora;
  - Sinistros devem ser criados a partir do método apropriado da classe Seguro;
  - CPFs e CNPJs devem ser verificados com os respectivos métodos.
- Ler arquivos de entrada de dados e fazer instanciação de objetos usando os dados dos arquivos;
- Apenas ao final da execução do projeto: chamar o menu de operações;
- Salvar as informações dos Objetos instanciados pelo sistema em um arquivos de saída contendo as informações relacionadas aos seguros e sinistros gerados.

id	dataInicio	dataFim	seguradora	listaSinistros	listaCondutores	valorMensal
1	2022-01-01	2022-12-31	Seguradora X	1,2,3	12345678901,23456789012,34567890123	100
2	2022-03-15	2023-03-14	Seguradora Y	4,5	45678901234,56789012345	150
3	2022-06-20	2023-06-19	Seguradora Z	6,7,8	67890123456,78901234567,89012345678,90123456789	200

Figura 2: Exemplo de saída de dados para informações sobre Seguro

id	data	endereço	condutor	seguro
1	2023-05-01	Rua A	75039491044	1001
2	2023-05-02	Rua B	42373028097	1002
3	2023-05-03	Rua C	66951213068	1003
4	2023-05-04	Rua D	11900860033	1004
5	2023-05-05	Rua E	67787629029	1005

Figura 3: Exemplo de saída de dados para informações sobre Sinistro

## 4 Observações

### 4.1 Implementação da Interface

A interface presente no diagrama define um método de gravar (*gravarArquivo*) e um de ler (*lerArquivo*) arquivos. Assim, as classes devem implementar as funções de leitura e escrita descritas pela interface de acordo com os seus respectivos construtores e arquivos de entrada e saída.

### 4.2 Métodos *lerDados()* e *gravarDados()* em Seguradora

O método *lerDados()* chama os métodos de leitura correspondentes à leitura que se deseja realizar, e esses métodos instanciam os objetos das classes correspondentes (por exemplo, *lerArquivo* da classe *ArquivoFrota* instancia objeto da classe *Frota*).

O método *gravarDados()*, por sua vez, chama os métodos de gravação correspondentes, e esses métodos geram arquivos de acordo com os dados que são passados (por exemplo, *gravarArquivo* da classe *ArquivoSeguro* recebe dados da classe *Seguro* - objeto).

Note que você pode definir os parâmetros desses métodos.

### 4.3 Classes que não estão no diagrama

Como pode-se notar o diagrama não contém certas classes exigidas em laboratórios anteriores. As funcionalidades dessas classes ainda estão sendo exigidas neste laboratório, porém a forma como elas estarão sendo implementadas fica a critério do aluno.

### 4.4 Entrada de Dados

Foram disponibilizados 5 arquivos CSVs contendo dados de Clientes, Condutores, Veículos e Frotas. O aluno deve usar esses arquivos para instanciar os respectivos objetos.

### 4.5 Saída de Dados

É esperado que o programa retorne dois arquivos no formato CSV. Um contendo as informações de Seguros (Figura 2) e um outro contendo as informações dos Sinistros (Figura 3).

## 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;
- Execução do código;
- Qualidade do código desenvolvido (saída dos dados na tela, tabulação, comentários);

- Instanciação dos objetos, e principais métodos das classes implementadas, na classe `AppMain`;
- Desenvolvimento correto dos métodos e classes requisitadas;

## 6 Entrega

- **A entrega do Laboratório é realizada exclusivamente via Github.** Para a submissão no Github, gere um release (tag) com a identificação do laboratório no estilo `<lab06-RA>`. Por exemplo, para o aluno com RA 123456, a tag será: `lab06-123456`.
- Observação: Evite criar releases enquanto não tiver certeza que seu código está funcionando como esperado.
- Utilize os horários de laboratório e atendimentos para tirar eventuais dúvidas de submissão e também relacionadas ao desenvolvimento do laboratório.
- **Prazo de Entrega:** 27/06 - 23h 59min

### 6.1 Organização das pastas do repositório

É esperado que seu repositório do Github contenha a mesma estrutura de pastas dos Laboratórios anteriores.