## Trabalho 2: Leitura de Arquivos e Data Access Objects

Rafael Guterres Jeffman 2 de abril de 2016

RGJEFFMAN@senacrs.com.br

## **Objetivos**

O objetivo deste trabalho é estudar a leitura de arquivos texto com formatação (arquivos CSV - Comma Separated Values) e criar uma estrutura de classes extensível para o estudo de diferentes estruturas de dados aplicadas à recuperação eficiente de dados. Também será trabalhada a ordenação dos dados através de algoritmos de ordenação simples.

## Instruções

Você deve implementar a estrutura de classes apresentada na Figura 1. Quando especificadas, as estruturas de dados a serem utilizadas não podem ser alteradas. Os algoritmos de ordenação utilizados devem conter o código fonte disponível na aplicação, e devem implementar o algoritmo especificado.

Você deve entregar apenas os códigos fontes (e a estrutura de diretórios, caso seja necessária). Serão aceitos apenas arquivos fontes, ou arquivos comprimidos com ZIP. Nenhum outro formato de compressão será aceito.

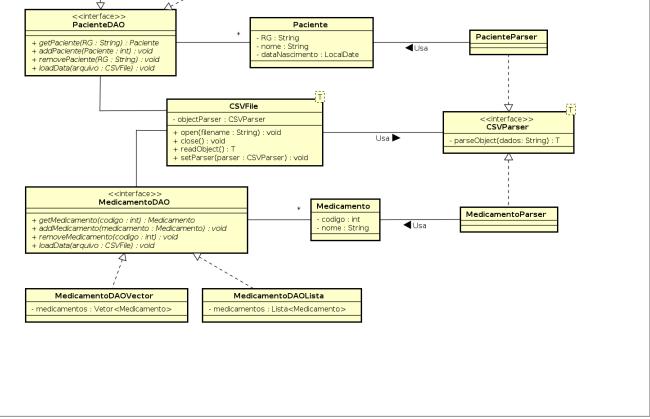
A aplicação deve funcionar da seguinte forma:

- 1. Ao iniciar a aplicação, o usuário seleciona que método de armazenamento deseja utilizar: Vetor ou Lista.
- Os arquivos de dados pacientes.csv e medicamentos.csv devem então ser carregados para as estrucuras MedicamentoDAO corretas (Lista ou Vetor).
- 3. A aplicação deve permitir a consulta de um paciente por RG ou por Nome.
- A aplicação deve permitir a consulta de um medicamento por Código ou por Nome.
- O paciente tem como "chave primária" o RG, o medicamento tem como chave primária o Código, logo, as estruturas devem estar ordenadas por estas chaves.

- 6. Para ordenar os dados, você deve utilizar algoritmos de ordenação, Selection Sort para o Vetor, e Insertion Sort para a Lista.
- 7. Para a leitura do arquivo, você deve criar a classe CSVFile, que poderá ser extendida para tipos de dados diferentes, dependendo do CSVParser utilizado. A responsabilidade dos objetos CSVParser é, dada uma linha do arquivo com os campos necessários, ele cria um novo objeto do tipo específico (Pessoa ou Medicamento).

## Tarefas Opcionais

- 1. Você pode permitir a criação de outros índices para as Listas e Vetores, por exemplo, com o **nome**. Nesse caso, crie métodos alternativos nos DAO, para retornar a nova lista ordenada.
- 2. Utilize apenas "Iteradores" para realizar a ordenação.



PacienteDAOLista

pacientes : ListaEncadeada<Paciente>

pkg

PacienteDAOVetor

pacientes : Vetor<Paciente>

powered by Astah