

## Atividade de PROG3 (Avaliação de Nota 1)

Com base no livro de referência da disciplina:

HORSTMANN, C. S.; CORNELL, G., Core Java Volume I - Fundamentos. 8ª edição. São Paulo: Pearson, 2009. (Ou versão mais recente). Códigos de exemplos em <https://horstmann.com/>

Obs.: Atividade inspirada nos assuntos tratados nos Capítulos 4 (Objetos e Classes) e 5 (Herança). Ou usar quaisquer referências compatíveis.

### > Descrição geral da atividade:

Considere a Figura 1 a visão detalhada das classes que compõem o ambiente de negócio de uma Clínica Hospitalar, que presta serviços de atendimento médico a pacientes, incluindo enfermagem para execução de exames radiológicos. Logo, você foi contratado para o desenvolvimento de um sistema Java com interface gráfica Desktop adequada para cadastro de Pacientes, Médicos e Enfermeiros, bem como o registro de Consulta Médica. Eis a seguir mais detalhamentos:

- **Prever construtores especializados para o cadastro de “Paciente”, “Médico” e “Enfermeiro”, observando a classificação de dados herdada a partir de “Dado Pessoal” de cada um.** Veja que todos devem ter obrigatoriamente o cadastro de “Endereço” e “Contato de Telefone/email”. O “Médico” e o “Enfermeiro” são atendentes do hospital, precisam ter cadastrados a CH semanal e o setor de trabalho, além de dados previstos em suas classes específicas. **Desenvolva convenientemente todas as classes indicadas na Figura 1 e crie métodos set/get para os atributos privados gerais, bem como métodos acessórios que se fizerem necessários.**

- **O cadastro de uma “Consulta Médica” envolve em identificar o “Paciente” e o “Médico” anteriormente cadastrados.** Assim, toda consulta estará vinculada à relação paciente-médico.

- **As classes “Endereço”, “Contato Telefone/email” e “Responsável” são acessórios aos cadastros gerais, bem como a classe “Gênero”, a ser construída como “enum” (*Enumerator*).**

- **Classe principal (main) .java** deverá concentrar os elementos de front-end (GUI para com o usuário), usando o GUI Designer disponível. Logo, esta tela principal deverá invocar as funcionalidades desejadas (que poderão gerar arquivos .java de interface específicos):

**I) cadastrar / consultar / editar / remover “pacientes”, “médicos” e “enfermeiros” cadastrados,** mantendo listas adequadas para cada categoria cadastrada em memória RAM (ArrayLists, p. ex.). Não misturar as categorias no filtro de informações para tal, criando assim telas cadastrais específicas.

**II) cadastrar / consultar / editar / remover “consultas médicas”,** tal como no item anterior.

**Dica:** Na interface, prever menus ou abas específicas para cada opção de cadastro de categoria desejado. Use tabelas para mostrar todas as informações cadastradas pertinentes a cada categoria. Separe as classes .java do front-end e do back-end utilizando “packages” diferentes em seu projeto. Veja que esta atividade espera que todas as classes da Figura 1 sejam implementadas.

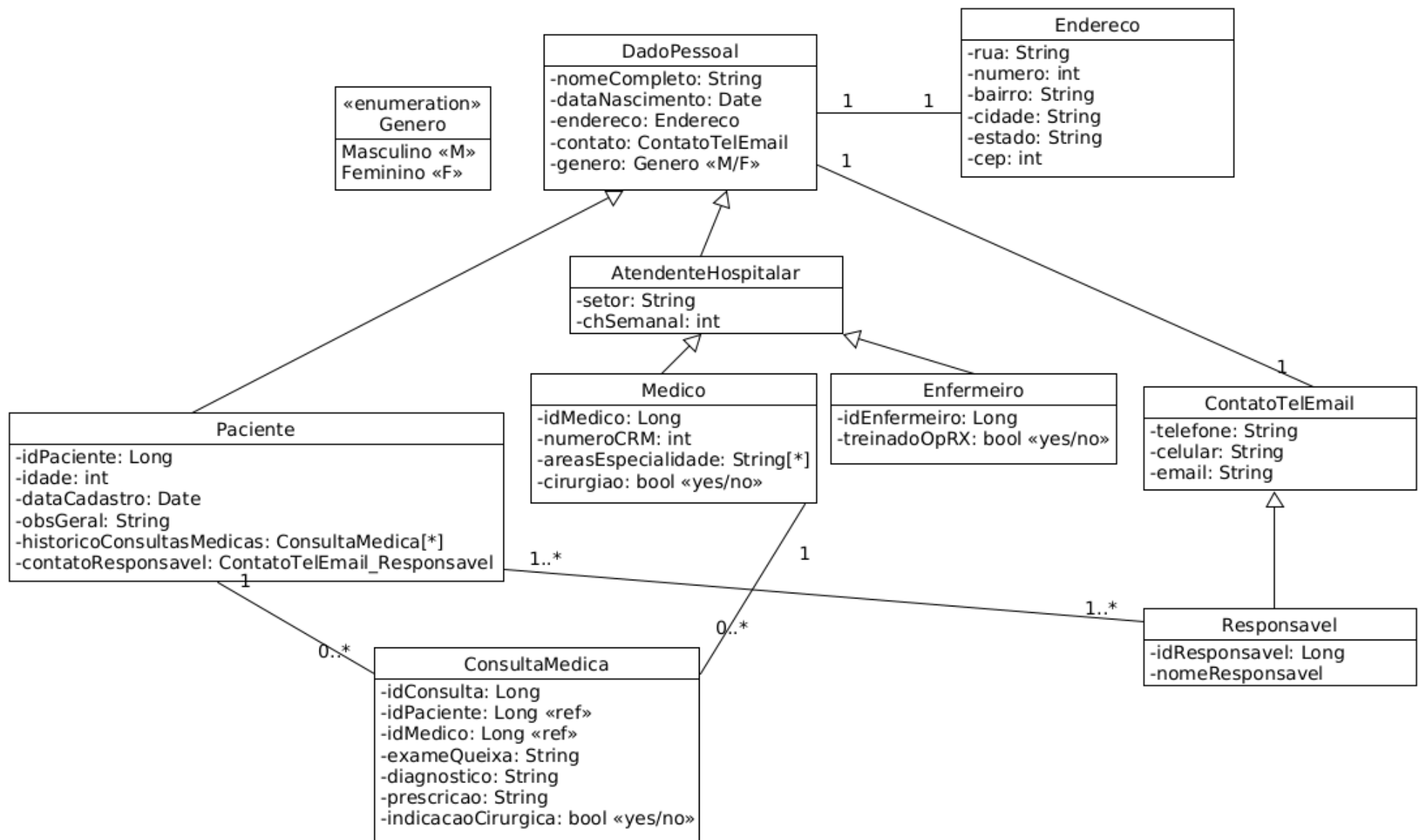


Figura 1 – Diagrama de Classes do ambiente de negócio de uma Clínica Hospitalar.

- **Criar uma classe “ExportarExcel”** especializada em exportar dados de cadastro via geração de um arquivo de saída, ao clicar em botão “Exportar Excel” por exemplo, salvando em arquivo de “nome\_escolido\_pelo\_usuario.xlsx” todos os dados cadastrados, organizados em 4 abas: “Paciente”, “Médico”, “Enfermeiro”, “Consulta-Médica”. **Dica:** Usar uma das diversas API Excel disponíveis na Internet, tais como: Apache POI, JExcel ou FastExcel. De mesmo modo, **criar uma segunda classe “ImportarExcel”**, para que se possa importar dados de cadastro a partir de um arquivo-padrão (gerado no passo anterior de exportação), atualizando assim toda a estrutura de dados em memória com tais dados.

**> Regras de documentação e submissão da atividade:**

**- Enviar ZIP contendo:**

1) Pasta do projeto Java completa;

2) README.DOCX descrevendo o desenvolvimento, dificuldades e testes do projeto (**colocar neste arquivo um hyperlink para vídeo público de simulação e explicação do projeto, para direcionamento da avaliação da atividade – obrigatório para considerar avaliação, dica: usar GDrive, p. ex.).**