Documentação Trabalho 1- POO

João Paulo da Cunha Faria

Divínopolis
Outubro de 2023

João Paulo da Cunha Faria

Documentação Trabalho 1- POO

Divínopolis
Outubro de 2023

0.1 Introdução

Este documento descreve o projeto e implementação de um sistema de gerenciamento de consultório médico solicitado por um consultório fictício. O sistema foi desenvolvido para armazenar informações sobre o consultório, pacientes, médicos e consultas marcadas. O sistema é destinado a melhorar a eficiência das operações do consultório, fornecendo uma maneira organizada de manter registros de pacientes e consultas.

0.2 Decisões de Implementação

Se tratando de um projeto que manipula dados de diferentes tipos e origens, assim como uma variada gama de entidades, **JAVA** foi linguagem de programaçao escolhida para programar o sistema que engloba a gestao de dados, mais especificamente versao 17.0.7.Por ser uma linguagem orientada a objeto **JAVA** se torna facilmente uma das melhores escolhas possiveis para realizar tal empreendimento.

0.2.1 Classe Pessoa

Inicialmente, por se tratar de um consultorio que realiza gestão de dados e de pessoas, a primeira decisão de implementação foi criar uma classe que simule e armazene dados de pessoas. Sendo assim, pesquisando a respeito dos dados com o qual o consultorio tem necessidade de lidar, no que diz respeito de seus pacientes e medicos ficou decidido que a melhor decisão, se tratando de dinamicidade e facilidade na manipulação do codigo, que as classes que manipulam dados relacionados a pessoas seriam divididos desta forma

1. Classe Pessoa que manipulara dados genericos relacionado ao cadastro de pessoas. Tendo seus atributos e metodos definidos da seguinte forma

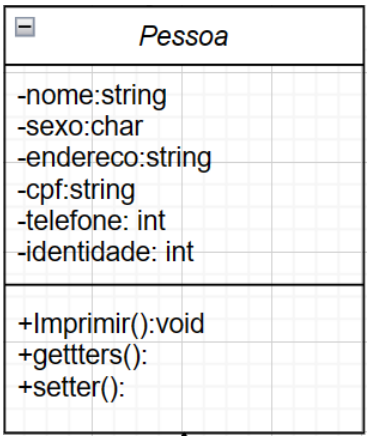

Figura 1 – Modelagem da classe Pessoa

2. Classe Paciente que lida apenas com dados diretamente ligados aos pacientes. Tendo em vista também que a classe Paciente herda todos os atributos da classe Pessoa, sendo assim a classe Paciente foi definida da seguinte forma.

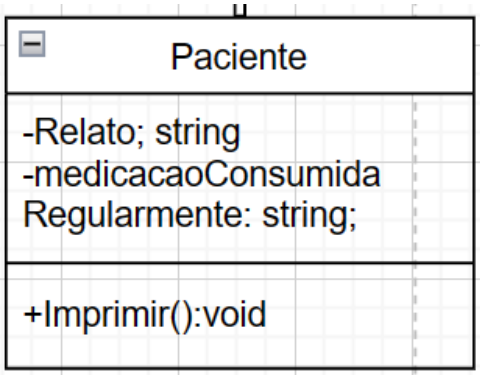

Figura 2 – Modelagem da classe Paciente

3. E por fim para abranger todos os dados relacionados a pessoas foi criado tambem a classe Medico, que por sua vez tambem herda todos os atributos da classe Pessoa. A definição final da classe medico se fez da seguinte maneira.

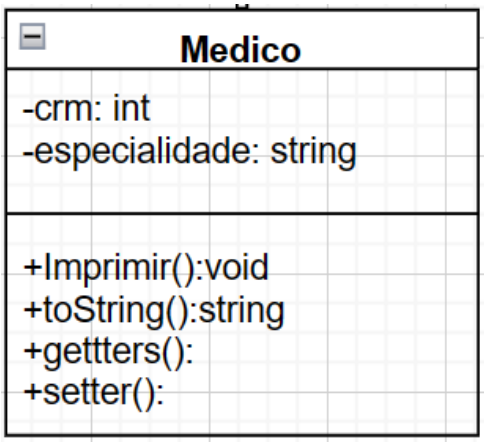

Figura 3 – Modelagem da classe Medico

4. O diagram final das classes que modelam dados relacionados a Pessoas

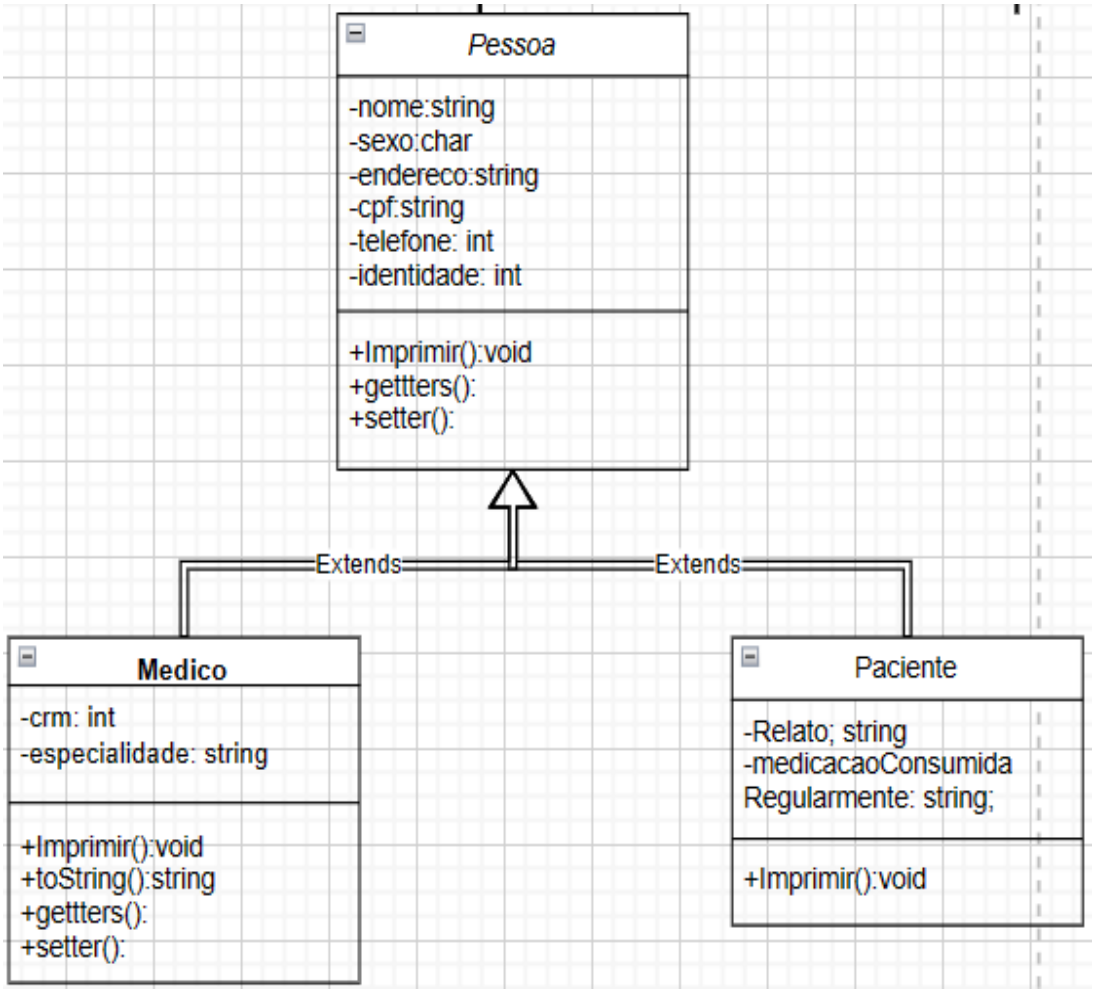

Figura 4 – Diagrama geral da classe Pessoa e suas heranças

0.2.2 Classe Consulta

Para modelar os serviços prestados pelo consultorio foi criado uma classe que relacione o medico que prestara o serviço e o paciente que sera atendido, assim como a hora e a data de quando sera realizado tal atendimento. A classe Consulta foi modelada da seguinte maneira, sendo uma classe independente e tendo apenas ralações com o consultorio.

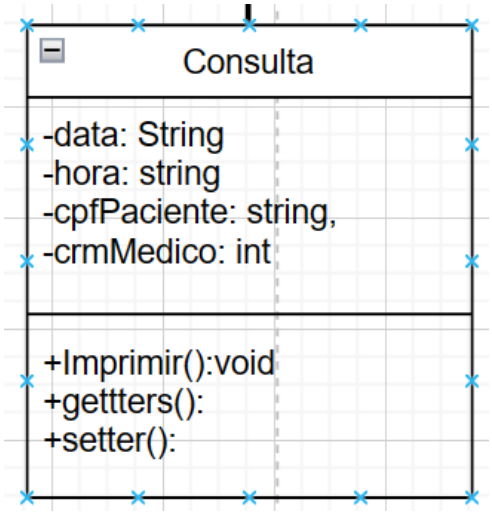

Figura 5 – Diagrama da classe consulta

0.2.3 Classe consultorio

Sendo a estrutra mais importante da implemtação a classe consultorio foi modelada de um modo que permita realacionar a ela todos os dados necessarios para a gestao do negocio. Seus atributos e metodos foram definidos da seguinte maneira.

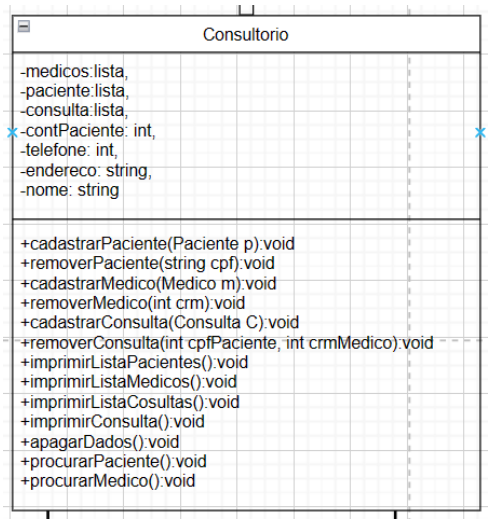

Figura 6 – Diagrama da classe consultorio

Tendo relacionametos com as classes paciente, medico, e consulta a classe Consultorio tem como objetivo relacionar, armazenar, apagar e cadastrar todas as informaçoes necessarias para a boa gestao do negocio.Por fim, seu diagrama de classe completo ficou modelado da seguinte forma.

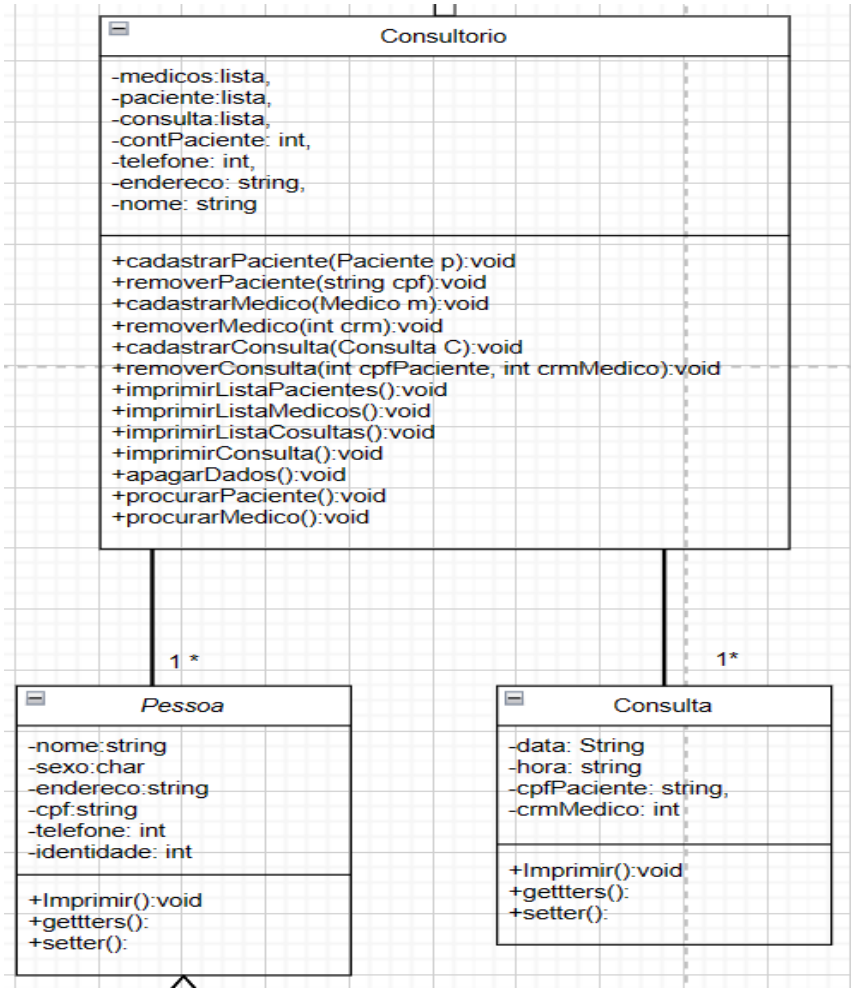

Figura 7 – Diagrama completo dos relacionamentos da classe Consultorio

0.2.4 Classe Metodos

Por ser um sistema que lida com uma grande quantidade de dados viu-se a necessidade da criação de uma classe que nos facilite pedir informações como nome, edereço, telefone... ao usuario de maneira dinamica havendo tambem reaproveitamento de codigo para casos onde seja precisso cadastrar novos pacientes ou medicos assim como em diversas outras situaçoes. Com o objetivo de fazer um codigo mais limpo e legivel a classes Metodos foi definida da seguinte forma.

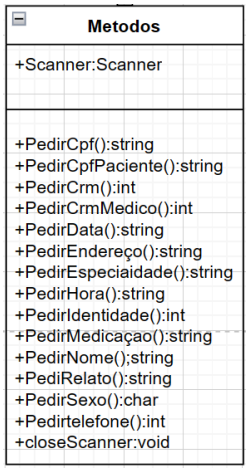

Figura 8 – Diagrama da classe Metodos

0.2.5 Classe Main-Teste

Para que seja possivel realizar testes em relaçao a implementaçao descrita, foi criada a classe main que nos permite utilizar todos os metodos e atributos. Tendo relações apenas com as classes Consultorio e Metodos sua modelagem foi definida da seguinte forma.

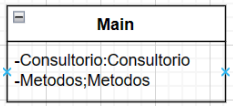

Figura 9 – Diagrama da classe Main

E por fim seu Diagrama geral modelando seus relacionamentos com as classes Consultorio e Metodos.

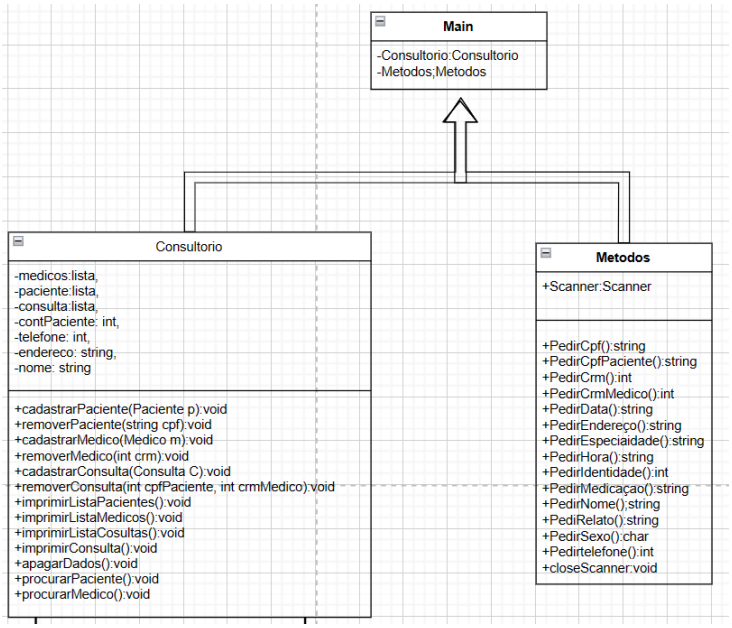

Figura 10 – Diagrama completo da classe Main

Todas as decisoes de implementaçao foram tomadas com o intuito de melhorar a eficiencia na gestao, facilitar a manutençao e leitura do codigo, e modelar da melhor maneira possivel manipulação de dados. Segue o diagrama completo da implementaçao.

Figura 11 – Diagrama completo da classe Main

0.3 Casos de teste

Afim de testar a eficiencia da implementaçao foram realizados testes abrangentes para garantir que o sistema funcione conforme o esperado. Casos de teste incluem:

1. Cadastro de médicos, pacientes e consultas.
2. Impressão de dados de médicos, pacientes e consultas.
3. Remoção de médicos, pacientes e consultas.
4. Manipulação de dados inválidos ou inexistentes.
5. Verificação de limites e capacidade de expansão das listas dinâmicas.

0.4 Resultados dos testes

1. Cadastro de Médicos, Pacientes e Consultas: O sistema permitiu o cadastro bem-sucedido de médicos, pacientes e consultas, capturando informações relevantes e armazenando-as de forma organizada.

2. Impressão de Dados: A funcionalidade de impressão de dados, tanto de médicos quanto de pacientes e consultas, forneceu resultados precisos e legíveis, tornando as informações facilmente acessíveis para consulta.

3. Remoção de Registros: O sistema permitiu a remoção adequada de médicos, pacientes e consultas, garantindo que os registros fossem excluídos corretamente das listas dinâmicas.

4. Tratamento de Dados Inválidos: O sistema demonstrou resiliência ao lidar com entradas inválidas ou incorretas, fornecendo feedback claro ao usuário e evitando falhas críticas.

5. Expansão das Listas Dinâmicas: Durante os testes, o sistema foi capaz de expandir dinamicamente as listas de médicos, pacientes e consultas à medida que novos registros eram adicionados, garantindo que não houvesse perda de dados.

6. Limites de Capacidade: Foram avaliados os limites de capacidade das listas dinâmicas para garantir que o sistema pudesse acomodar um número significativo de registros sem problemas de desempenho ou alocação de memória.

7. Operações de Consulta: O sistema respondeu eficientemente a operações de consulta, como a busca por informações de pacientes ou médicos específicos, exibindo resultados precisos.

8. Feedback ao Usuário: O sistema forneceu feedback claro e informativo ao usuário, orientando-o nas operações e ajudando-o a entender as ações realizadas.

0.5 Análise dos resultados

O sistema desenvolvido atende às necessidades do consultório médico, proporcionando um meio eficaz de armazenar e gerenciar informações de pacientes, médicos e consultas. Ele é capaz de lidar com consultas do usuário de forma apropriada e fornece feedback adequado em caso de erros ou operações bem-sucedidas.

Qualquer melhoria futura ou expansão do sistema pode ser considerada, mas a estrutura atual oferece uma base sólida para o gerenciamento de consultas médicas e informações relacionadas.