Trabalho 1

Linguagem de Programação Orientada a Objetos

P03- 2020-1

## **1. Integrantes do Grupo**

**Nome:** Ana Carolina de Araújo Leite **RGA:** 2019.1902.012-8

**Nome:** Marcus Vinicius Romero Nobre Leal **RGA:** 2017.1906.120-3

**Nome:** Guilherme Moreno Guimarães **RGA:** 2019.1902.024-1

**Nome:** Julio Cezar Rodrigues de Souza **RGA:** 2019.1902.025-0

**Nome:** Giulia Caroline Giorgini **RGA:** 2019.1902.029-2

**2. Proposta de Problema e Solução**

-Não deixou explícitas as funcionalidades que exigem

polimorfismo. “Específica os métodos que vão utilizar polimorfismo”

- No diagrama, não diferenciou as relações de

associação ou herança entre as classes.

- As responsabilidades das classes estão estranhas.

Por exemplo, não faz muito sentido a classe Atendente

ser responsável pelo método imprimeProntuario().

Deveria haver uma classe Prontuario que armazenaria

os atributos de um prontuário. A classe que armazena

os prontuários poderia ter métodos de relatório.

- Também não está claro as funcionalidades do

sistema em si, do ponto de vista do usuário. OK

**Proposta do problema**

O diretor-geral do Hospital Boa Saúde está com uma dificuldade em organizar os prontuários dos pacientes do ambulatório. Ele quer um software que o auxilie na organização das informações de seus funcionários e pacientes. Para o diretor-geral o software deverá:

1. Cadastrar um novo paciente.
2. Classificar a prioridade de atendimento do paciente de acordo com os sintomas apresentados.
3. Organizar os prontuários dos pacientes por gravidade dos sintomas.
4. Cadastrar seus funcionários.
5. Cadastrar as doenças mais frequentes do Hospital Boa Saúde.
6. Imprimir um relatório com todos os pacientes que deram entrada em determinado dia.
7. Imprimir um relatório mensal com todas as doenças diagnosticadas no Hospital Boa Saúde.

**Solução a ser desenvolvida**

**Polimorfismo**

“Permite que um mesmo nome represente vários comportamentos diferentes.”

um objeto nasce de um tipo e morre daquele tipo, o que pode mudar é a maneira como nos referimos a ele

definição que eu entendi: são os métodos das classes que se comportam de maneira diferente na classe filha.

**Herança**

O relacionamento entre Funcionário e Médico é de Herança

“A herança será aplicada tanto para características quanto pra os comportamentos, toda classe filho tem as características e comportamentos herdados da mãe.”

1. Conter uma classe Pessoa com os atributos nome(String), cpf(String), sexo(char).
2. Conter como subclasse de Pessoa uma classe Funcionário que além dos atributos da classe Pessoa contenha o atributo idFuncionario(int).
3. Conter como subclasse de Funcionário uma classe Atendente que terá um método imprimeProntuario() que vai imprimir o relatório.
4. Conter como subclasse de Funcionário uma classe Médico que além dos atributos da classe Funcionário herdados contenha o atributo numeroRegistro(int).
5. Conter como subclasse de Funcionário uma classe Atendente que deve possuir um método imprimeProntuario(), esse método irá gerar um prontuário que contenha: o nome do paciente(String), o cpf(String), a enfermidade(String), a enfermeira responsável(String), o médico responsável(String), a data de entrada(tipoData) e a data de saída(se já houver saído) e o convênio(String). Ela também deve possuir um método consultaCadastro(boolean) em que poderá verificar se o paciente já possui cadastro no sistema.
6. Conter como subclasse de Paciente e uma classe Doença deve herdar o atributo sintomas(String) e possuir o atributo nome(String) e sua classificação Internacional de Doenças cid(String)

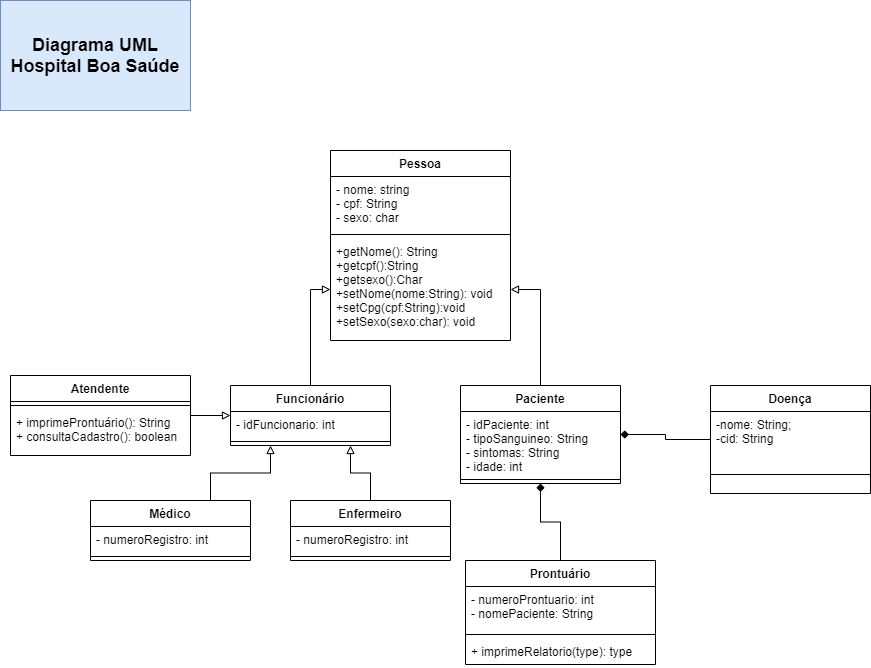
****

Diagrama UML para o software do Hospital Boa Saúde

\*\* armazenar em prontuario as informações recebidas de paciente