

CENTRO UNIVERSITÁRIO SENAC

Emerson de Oliveira Barbosa
Marlon de Oliveira Barbosa
Paulo Eduardo da Silva Filho
Rafael Ribeiro de Oliveira
Rudibert Baade

**Projeto Integrador 3: desenvolvimento de sistemas orientados
a objetos**

SÃO PAULO
2021

CENTRO UNIVERSITÁRIO SENAC

Emerson de Oliveira Barbosa
Marlon de Oliveira Barbosa
Paulo Eduardo da Silva Filho
Rafael Ribeiro de Oliveira
Rudibert Baade

Trabalho de Projeto Integrador 3
do Centro Universitário SENAC
Campus Santo Amaro.

Orientador Profº Luiz Bono

SÃO PAULO
2021

LISTA DE ILUSTRAÇÕES

Figura 1 – diagrama de classes	7
Figura 2 – diagrama de objetos	8

SUMÁRIO

1 INTRODUÇÃO	5
2 DIAGRAMA DE CLASSES	6
3 DIAGRAMA DE OBJETOS	8
REFERÊNCIAS	9

1 Introdução

O presente projeto tem como objetivo a criação de dois diagramas da linguagem UML (*Unified Modeling Language*), quais sejam: diagrama de classes e diagrama de objetos. Posteriormente, os mesmos serão implementados via linguagem Java, sendo que, para isso, todo o código-fonte será inserido no final do presente trabalho.

2 Diagrama de classes

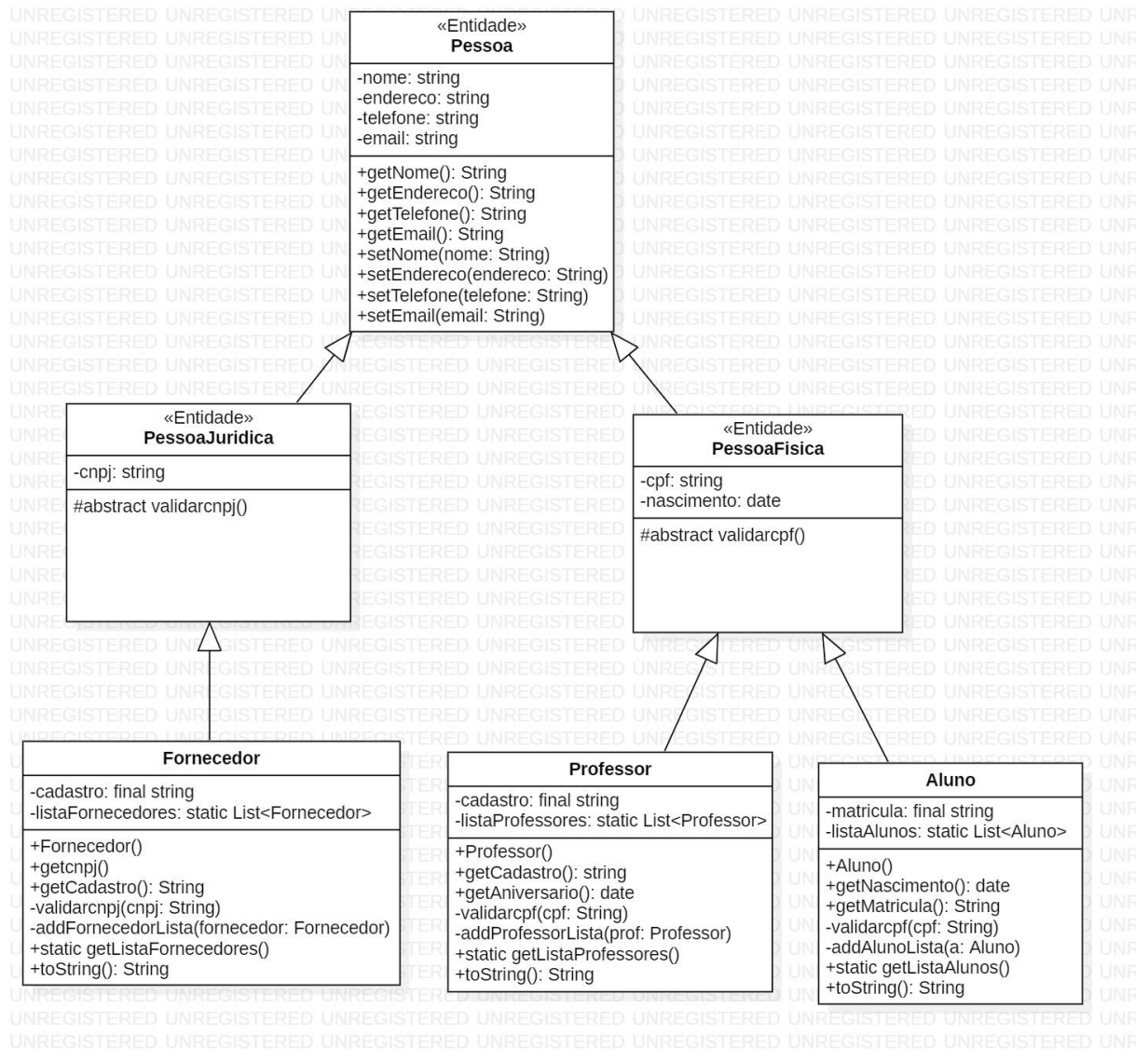
O projeto propõe a criação de cinco entidades para um sistema a ser utilizado em ambiente acadêmico. Para garantir uma relação adequada entre as classes, foi implementado no diagrama o princípio da herança. Dessa forma, os atributos e os métodos presentes em *PessoaJuridica* serão herdados pelo *Fornecedor*, e em *PessoaFisica* pelo *Professor* e *Aluno*.

Vale ressaltar que, com o intuito de relacionar todas as classes, também foi necessária a definição de uma sexta classe “Pessoa” que detém todos os atributos e métodos considerados mais básicos e que fazem parte de todas as classes filhas, como nome, endereço, email e telefone.

No que se refere a implantação dos métodos que retornem listas sobre os fornecedores, professores e alunos cadastrados no sistema, a solução proposta foi a declaração, em cada das classes filhas, de uma lista *static*. Além disso, foram implementados métodos privados que realizem a inclusão de cada nova instância dentro dessas listas estáticas.

Contudo, para que funcione, tais operações precisam ser chamadas dentro dos construtores a fim de que, quando um novo objeto seja instanciado, automaticamente, os métodos façam as devidas inclusões nas variáveis de classe. Na perspectiva do usuário, a chamada dessas listas poderão ser feitas por meio das operações estáticas *getListaFornecedores()*, *getListaProfessores()* e *getListaAlunos()* sem a necessidade de instanciação de objetos.

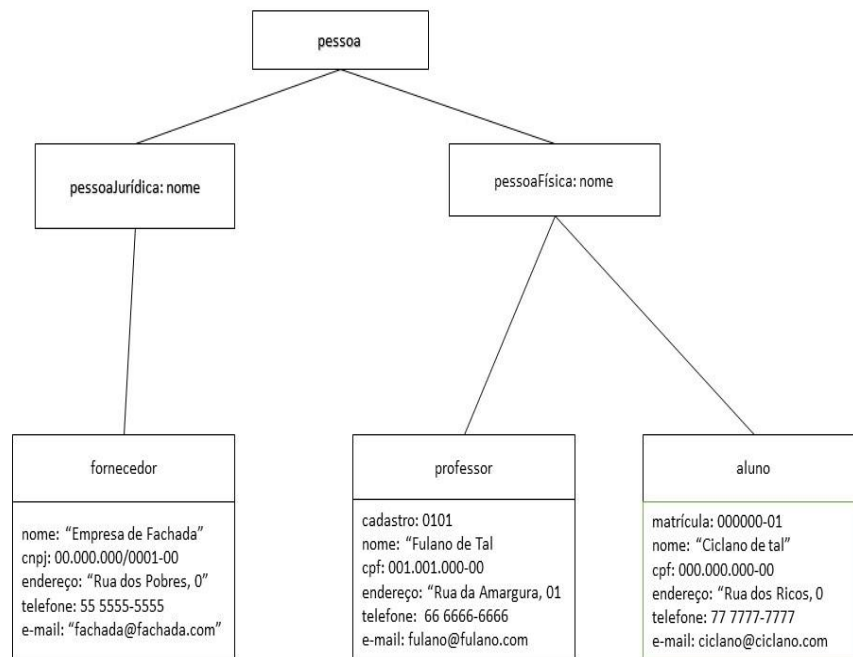
Por último, como o sistema lidará com CNPJ e CPF, foram criados métodos de validação dessas informações. Tais operações foram inseridas nas classes abstratas *PessoaJuridica* e *PessoaFisica* de modo estático para que, obrigatoriamente, sejam implementadas nas classes filhas.



3 Diagrama de objetos

A fim de representar como ficará a instanciação de objetos utilizando as classes já representadas anteriormente, o diagrama de objetos desenhado é composto pela hierarquia entre elas.

As classes Pessoa, PessoaJuridica e PessoaFisica não foram instanciadas, visto que são abstratas, cabendo apenas às classes mais abaixo a geração de todos os atributos e métodos obtidos via herança. Isso evita eventuais problemas decorrentes da instanciação equivocada pelo usuário das classes-pai, mantendo, inclusive, a integridade dos dados inseridos e uma usabilidade mais consistente.



REFERÊNCIAS

Oberleitner, Allen. **Programação orientada a objetos** / Allen Oberleitner, Andrey Araujo Masiero. São Paulo: Editora Senac São Paulo, 2021. (Série Universitária)