

Objetivo 3: Associação Binária, Agregação, Composição, e outras associações entre classes

Lista 1

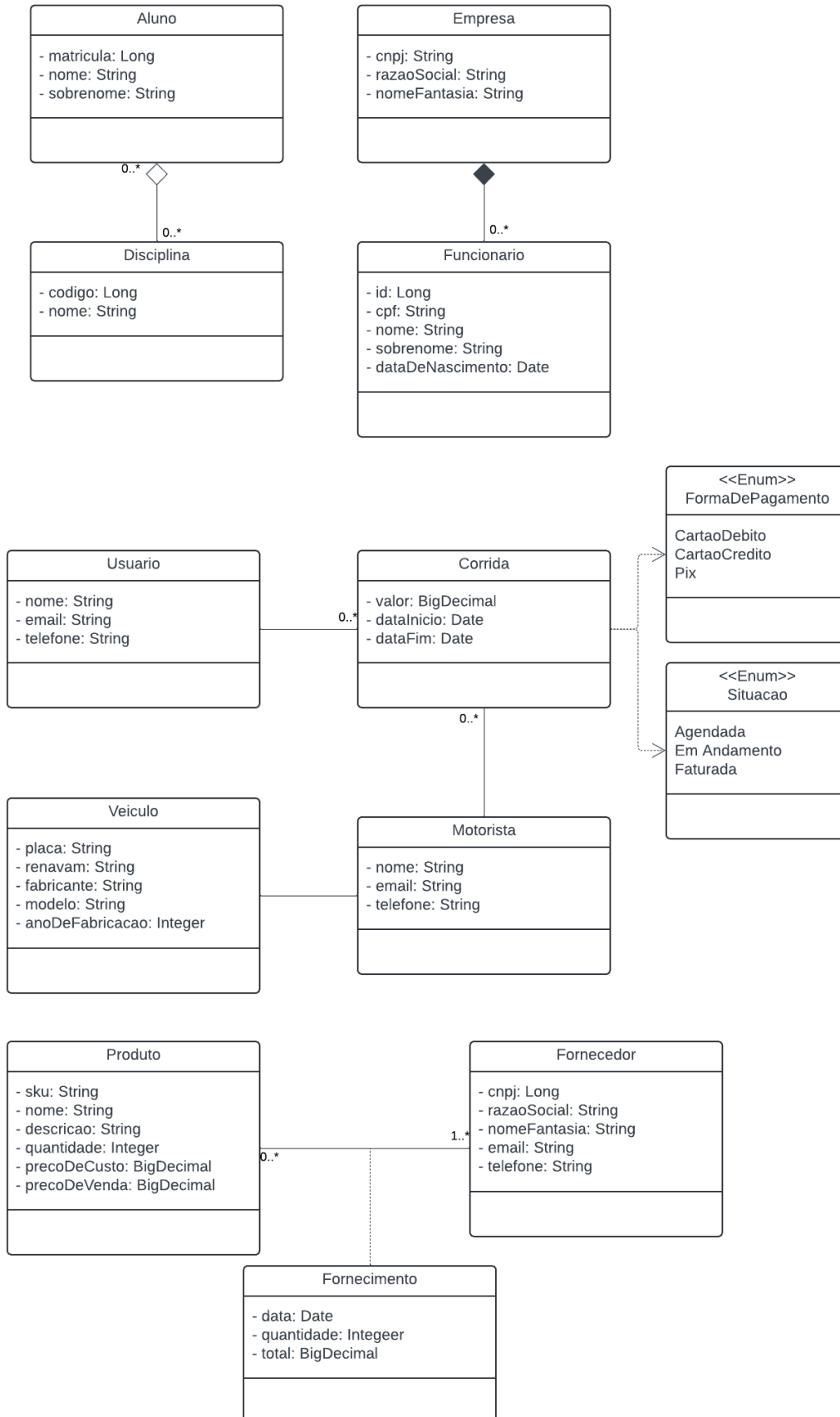
Objetivos

1. Assimilar os conceitos de outras associações entre classes, como, agregação, composição, associação binária, e classe associativa;
2. Entender a relação entre objeto-todo e objeto-parte;
3. Entender como programar as relações de agregação e composição;
4. Entender como programar as relações binárias;
5. Entender como programar as classes associativas;
6. Interpretar diagramas de classe da UML para transformá-los em código na tecnologia Java;
7. Criar pequenos apps Java em modo texto.

EXERCÍCIO

1. Faça o que se pede:
 - a. Crie um projeto Java na arquitetura de software MVC;
 - b. Interprete os diagramas de classes apresentados na Figura anexa (página 2);
 - c. Para o diagrama Aluno-Disciplina:
 - a. Crie dois objetos de cada classe, e insira valores válidos em seus atributos;
 - b. Matricule cada um dos alunos nas duas disciplinas.
 - c. Imprima o relatório de disciplinas por aluno, em ordem crescente, critério nome da disciplina.
 - d. Para o diagrama Empresa-Funcionário:
 - a. Crie um objeto da classe Empresa e dois objetos da classe Funcionário, e insira valores válidos em seus atributos;
 - b. Adicione todos os funcionários na empresa;

- c. Imprima a lista de funcionários em ordem crescente, critério nome.
- e. Para o diagrama Usuário-Corrída-Motorista-Veículo (uma espécie de Uber):
 - a. Faça o programa realizar duas corridas para o mesmo usuário, com um mesmo motorista e veículo;
 - b. Faça o programa imprimir todas as **corridas por usuário**, em ordem decrescente, pelo critério valor da corrida;
 - c. Faça o programa imprimir todas as **corridas por motorista**, em ordem crescente, pelo critério data de início da corrida;
 - d. Faça o programa imprimir o valor **total de todas as corridas** realizadas por motorista, e imprima os dados do motorista e o valor calculado.
- f. Para o diagrama Produto-Fornecedor-Fornecimento:
 - a. Faça o programa registrar dois fornecimentos, cada fornecimento para um produto diferente;
 - b. Imprima o relatório de fornecimentos (todos os fornecimentos realizados), contendo, os dados do produto, os dados do fornecedor, o total de cada fornecimento, e o total de todos os fornecimentos;



Gabarito

Você encontra o gabarito deste exercício no Github, em:

https://github.com/vagnersilvaifsul/lpoo/tree/main/objetivo3_lista1/src

Referências Bibliográficas

JAVA 17. **JDK 17 Documentation**. Disponível em: <https://docs.oracle.com/en/java/javase/17/>. Acesso fevereiro, 2024.

JAVA 17 API. **Java Platform, Standard Edition & Java Development Kit Version 17 API Specification**. Disponível em: <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>. Acesso fevereiro, 2024.

DEITEL, Paul J.; DEITEL, Harvey M. **Java: como programar**. 8. ed. São Paulo: Pearson, 2010.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. 2. ed. São Paulo: Novatec, 2011.