**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA**

**UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE**

**Bacharelado em Engenharia de Software**

**Filipe Iannarelli Caldeira**

**Gabriel Vinicius Ramos da Silva**

**Paulo Angelo Dias Barbosa**

**Wesley Mouraria Pereira**

**TRABALHO PRÁTICO – DAO (*DATA ACCESS OBJECT*)**

Belo Horizonte

2019

**Filipe Iannarelli Caldeira**

**Gabriel Vinicius Ramos da Silva**

**Paulo Angelo Dias Barbosa**

**Wesley Mouraria Pereira**

**TRABALHO PRÁTICO – DAO (*DATA ACCESS OBJECT*)**

Trabalho de Software apresentado como requisito parcial à aprovação na disciplina Programação Modular

Professores:

Hugo Bastos de Paula

Belo Horizonte

2019

**SUMÁRIO**

[1. Objetivo 4](#_Toc9178749)

[2. Introdução 4](#_Toc9178750)

[3. Classes 6](#_Toc9178751)

[3.1. Cliente 6](#_Toc9178752)

[3.2. ClienteDAO 6](#_Toc9178753)

[3.3. Evento 7](#_Toc9178754)

[3.4. EventoDAO 7](#_Toc9178755)

[3.5. DAO 7](#_Toc9178756)

[3.6. Teste 7](#_Toc9178757)

[3.6. TesteClienteDAO e TesteEventoDAO 7](#_Toc9178758)

[4. Anexos 8](#_Toc9178759)

# 1. Objetivo

O objetivo deste trabalho é apresentar um modelo de persistência de dados baseado no padrão de projeto denominado DAO (Data Access Object), utilizando a linguagem Java e os conceitos de programação orientada a objetos, tendo em consideração conceitos de programação modular.

# 2. Introdução

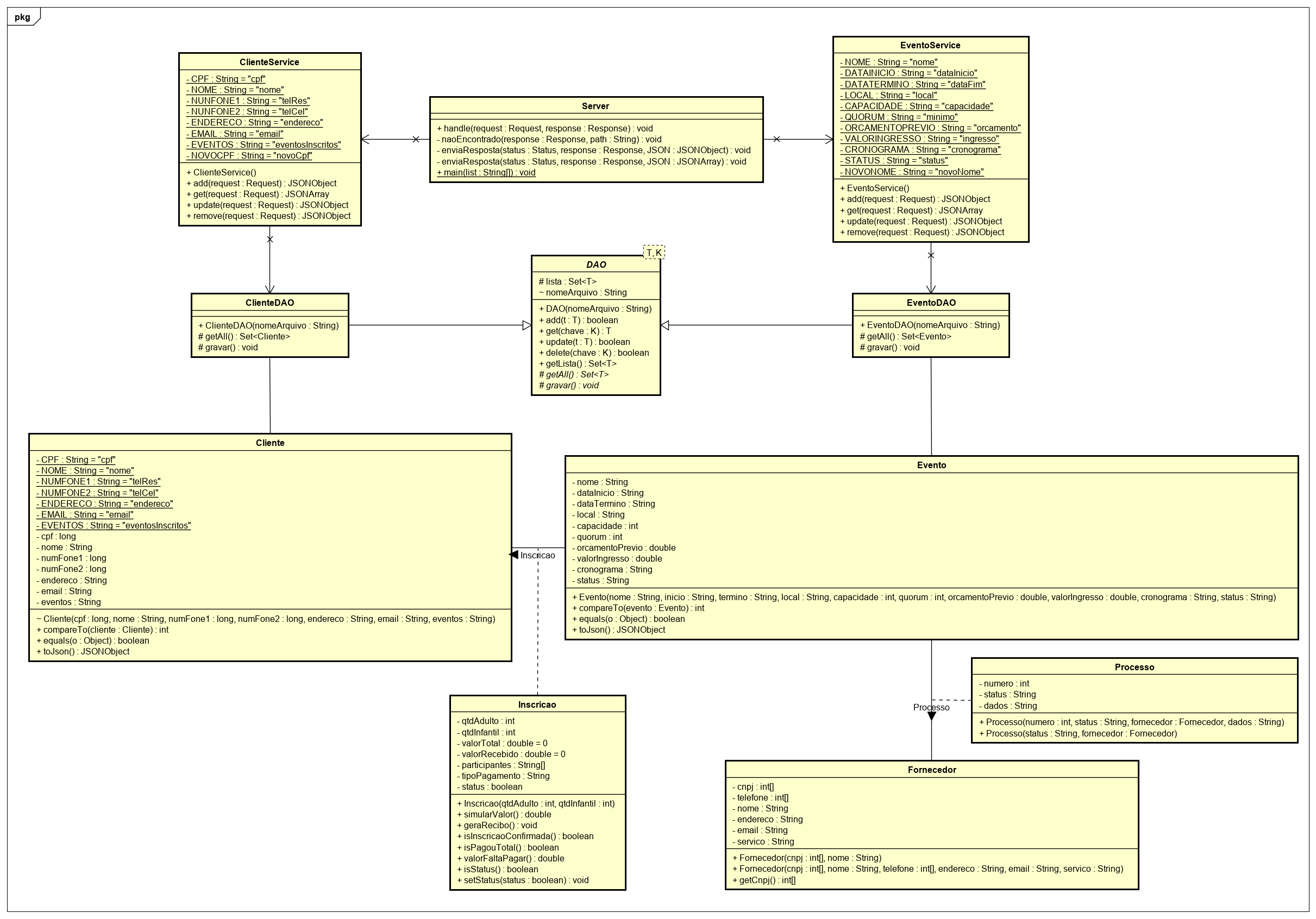
Para a elaboração do programa objeto deste trabalho, foi utilizado o mesmo problema para o trabalho da disciplina de TIS II (Trabalho Interdisciplinar de Software II), ainda em desenvolvimento.

A solução aborda o desenvolvimento de uma aplicação para auxiliar no gerenciamento de excursões de escalada para a empresa Summit Ginásio de Escalada.

Até o momento, foram desenvolvidas funcionalidades de CRUD (create, read, update, delete – ou criar, ler, atualizar, remover) para duas categorias: clientes e eventos, sendo que este trabalho se limitará a estas funcionalidades.

O diagrama de classes do programa em questão é o apresentado a seguir. Por se tratar de trabalho integrado com o da disciplina de TIS II, tal diagrama apresenta classes que ainda não foram desenvolvidas no trabalho de TIS, sendo elas: Inscrição, Fornecedor, Processo.

**Figura 1 – Diagrama de Classes**

****

**Fonte: elaborado pelos autores**

# 3. Classes

Para a aplicação desenvolvida, foram criadas as seguintes classes:

- Cliente

- ClienteDAO

- Evento

- EventoDAO

- DAO

- Teste

- TesteClienteDAO

- TesteEventoDAO

## 3.1. Cliente

Esta classe é responsável pelos objetos do tipo Cliente, ou seja, ela armazena todas as informações pertinentes a um cliente (através dos atributos) e possui os métodos necessários para a manipulação desses objetos, como, por exemplo, o construtor e os *setters* e *getters.*

Como ela segue o padrão da interface Comparable, foi inserido um método de implementação *compareTo()*, responsável por como essa classe se comporta em relação à interface implementada.

## 3.2. ClienteDAO

Responsável pela persistência de dados, a classe ClienteDAO faz a manipulação do arquivo de registro de clientes.

Por estender da classe DAO, possui os métodos de CRUD por herança, todos implantados na classe DAO.

## 3.3. Evento

Análoga à classe Cliente, a classe Evento armazena as informações de eventos e possui os métodos necessários para a manipulação desses objetos.

Como ela também segue o padrão da interface Comparable, também possui o método *compareTo()*.

## 3.4. EventoDAO

Da mesma forma que a classe Evento é análoga à classe Cliente, a classe EventoDAO é de ClienteDAO, sendo responsável pela persistência de registros de eventos em arquivos.

Também estende da classe DAO, possuindo métodos de CRUD por herança.

## 3.5. DAO

Classe abstrata que possui os atributos e métodos necessários a suas classes herdeiras, foi criada para permitir que as classes DAO de diferentes tipos de objetos possam trabalhar de maneira polimórfica.

## 3.6. Teste

Classe de execução do programa, onde está contido o método *main()*, é por onde o programa inicia e executa as demais funcionalidades quando necessário.

## 3.7. TesteClienteDAO e TesteEventoDAO

Estas duas classes são classes de teste, responsáveis pela verificação das funcionalidades de persistência do programa, se tais funcionalidades estão operando corretamente.

# 4. Anexos

Os arquivos de código do programa, em linguagem Java, e o diagrama de classes são anexos deste documento.