

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE

BRUNO LINS DOS SANTOS VIANA E RÔMULO DA SILVA CAVALCANTI

IMPLEMENTAÇÃO DO SISTEMA: Academia muita bomba

DESCRIÇÃO DO SISTEMA A SER DESENVOLVIDO

O atual trabalho consiste na etapa de implementação de um sistema dedicado a uma academia fictícia denominada: Academia Muita Bomba. O objetivo da dupla foi criar um sistema com funcionalidades que fizessem sentido para uma academia como uma matrícula contendo informações básicas do cliente e uma forma de promover e organizar aulas realizadas pela empresa.

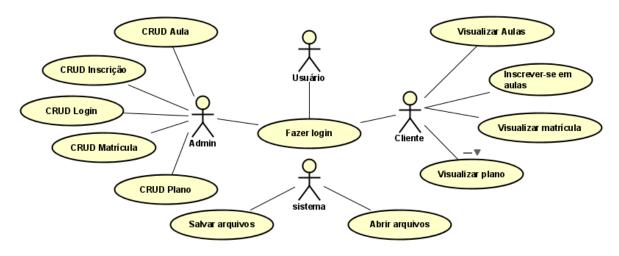
O sistema contém duas visões diferentes dependendo do tipo de permissão oferecido por cada conta registrada nele. A princípio, a dupla dividiu as contas em 3, mas tornou-se evidente que um dos tipos, o personal, era bastante similar ao admin e, por isso, o grupo reduziu o número de visões a duas, uma de admin e outra de cliente.

O tipo admin tem permissão para controlar todo o sistema sendo possível criar, ler ou listar, atualizar e deletar (CRUD) aulas, inscrições, logins, matrículas e planos. O cliente possui o menor nível de permissão e, portanto, está limitado apenas a visualizar e realizar inscrições em aulas, visualizar seu plano e verificar as informações de sua matrícula.

Dessa forma, listando as funcionalidades esperadas do sistema temos: criação, listagem, atualização e remoção de aulas, inscrições, logins, matrículas e planos; inscrição em aulas e visualização a certas informações restritas a conta.

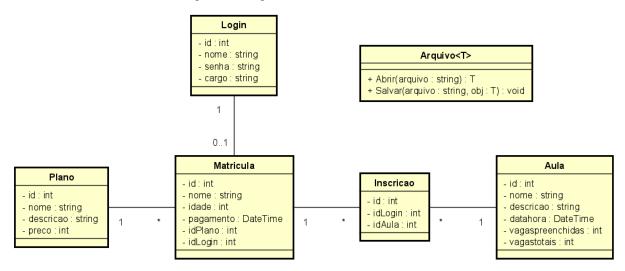
O repositório contendo todos os arquivos utilizados na produção do atual projeto pode ser acessado através do link: https://github.com/BrunoLTads/Academia-Muita-Bomba.

Figura 1. Diagrama de casos de uso



Fonte: Autores (2023).

Figura 2. Diagrama de classes de entidades



Fonte: Autores (2023).

Figura 3. Diagrama de classe da camada de negócio contendo as operações do sistema

Camada de Negócios

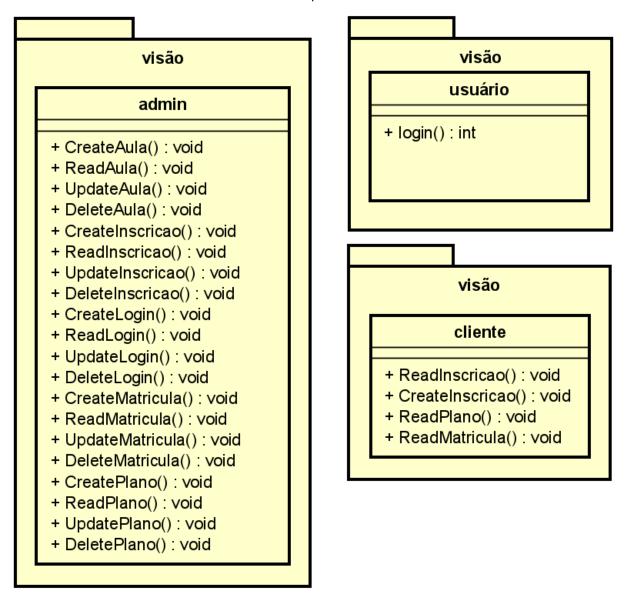
<<static>>

Sistema

- matriculas : List<Matricula>
- aulas : List<Aula>logins : List<Login>
- planos : List<Plano>
- inscrições : List<Inscrição>
- + LoginCreate(obj : Login) : void
- + LoginRead(): List<Login>
- + LoginRead(id : int) : Login
- + LoginUpdate(obj : Login) : void
- + LoginDelete(obj : Login) : void
- + MatriculaCreate(obj : Matricula) : void
- + MatriculaRead(): List<Matricula>
- + MatriculaRead(id : int) : Matricula
- + MatriculaUpdate(obj : Matricula) : void
- + MatriculaDelete(obj : Matricula) : void
- + AulaCreate(obj : Aula) : void
- + AulaRead(): List< Aula>
- + AulaRead(id:int): Aula
- + AulaUpdate(obj : Aula) : void
- + AulaDelete(obj : Aula) : void
- + PlanoCreate(ob j : Plano) : void
- + PlanoRead(): List<Plano>
- + PlanoRead(id: int): Plano
- + PlanoUpdate(obj : Plano) : void
- + PlanoDelete(obj : Plano) : void
- · Franciberete(obj. Franci) . Volu
- + InscricaoInserir(ob : Inscricao) : void
- + InscricaoRead(): List<Inscricao>
- + InscricaoRead(id : int) : Inscricao
- + InscricaoUpdate(obj : Inscricao) : void
- + InscricaoDelete(obj : Inscricao) : void
- + ArquivosAbrir(): void
- + ArquivosSalvar(): void

Fonte: autores (2023).

Figura 4. diagrama de classes da interface com o usuário contento todas as operações que podem ser chamadas pelos usuários.



Fonte: autores (2023).