



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

PROGRAMAÇÃO ORIENTADA A OBJETOS

*1º Relatório do Projeto **Sistema de Gerenciamento de Academias (SGA)***

Desenvolvedor: Pedro Henrique **Magalhães Botelho**

Desenvolvedor: Daniel Vitor **Pereira Rodrigues**

Orientador: Atílio **Gomes Luiz**

GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO - 6º SEMESTRE

QUIXADÁ - CE
2022

Conteúdo

1	Introdução	1
2	Descrição do Projeto	2
3	Recursos do Sistema	2
4	Planejamento	3
4.1	Diagrama UML	4
5	Conclusão	5

1 Introdução

Partindo da proposta para a disciplina de **Programação Orientada a Objetos**, pelo professor Atílio G. Luiz, é apresentado aqui a idealização do projeto do **Sistema de Gerenciamento de Academias**, o SGA, como o projeto final da disciplina.

O projeto partiu de uma proposta mais generalizada e fluiu em uma análise mais específica: construir um sistema simples, porém robusto, que atenda as necessidades de uma pequena academia. Partindo de uma visão simplista uma academia tem alguns pontos que um gerenciamento mais efetivo deve ser usado, tais como: manejar seus alunos, seus funcionários e sua caixa, gerenciando corretamente as mensalidades dos alunos e os salários dos funcionários.

Projeto: Sistema de Gerenciamento de Academias (SGA)

Instituição: Universidade Federal do Ceará (UFC)

Disciplina: Programação Orientada a Objetos (POO)

Desenvolvedores: Pedro Botelho & Daniel Vitor

Orientador: Atílio Gomes

2 Descrição do Projeto

O projeto foca em desenvolver um sistema simples e prático para gerenciamento de pequenas academias, estabelecendo uma relação de simplicidade e robustez. O sistema, a priori, irá realizar o controle do cadastro e dos caixa de alunos e funcionários, utilizando um sistema de banco de dados em arquivo para assegurar a persistência de dados.

Sendo desenvolvido em Java, o sistema, a priori, se utilizará de uma interface de linha-de-comando, também conhecida como CLI, entregando uma interface fácil de usar para o usuário.

O sistema deverá ser encapsulado em um executável .JAR, de forma que fique mais fácil a inicialização do sistema pelo cliente, sendo necessário apenas clicar duas vezes no lançador, e, como não será utilizado um sistema de banco de dados com servidor local, como MySQL, também não será necessário realizar mais passos.

O sistema tem, atualmente, 6 classes concretas, sem contar a classe principal da aplicação e os enums, que modelam algo dentro do sistema, tal como a empresa da academia, alunos, funcionários, e os gerenciadores. Falaremos em breve sobre eles.

Os alunos, bem como os funcionários e o próprio banco de dados, serão gerenciados por classes próprias, as classes gerenciadoras, que irão manejar a folha de pagamento e as mensalidades, bem como manter os conjuntos de funcionários e alunos, de forma a construir um banco de dados da empresa. A empresa é modelada por uma classe especial, que guarda as principais informações da academia.

O sistema ainda conta com um mecanismo de mensalidade sem folga. Isso quer dizer que a situação do aluno irá ficar "em atraso" assim que chegar o dia de sua mensalidade vencer.

3 Recursos do Sistema

O sistema deverá ter os recursos essenciais para realizar as tarefas básicas da parte administrativa de uma academia, tais como gerenciar alunos e o caixa.

Vejamos alguns recursos do sistema:

- Modelagem da empresa e Gerenciador de Caixa
- Cadastro e gerenciamento de Alunos
- Cadastro e gerenciamento de Funcionário, com cargos especificados por um Enum
- Sistema de Mensalidade sem Folga
- Gerenciamento de Banco de Dados local

4 Planejamento

Como já dito, o sistema foi desenvolvido em Java. Já o diagrama UML foi totalmente projetado em PlantUML, pois oferece uma interface muito robusta e simples para se desenhar diagramas UML via código.

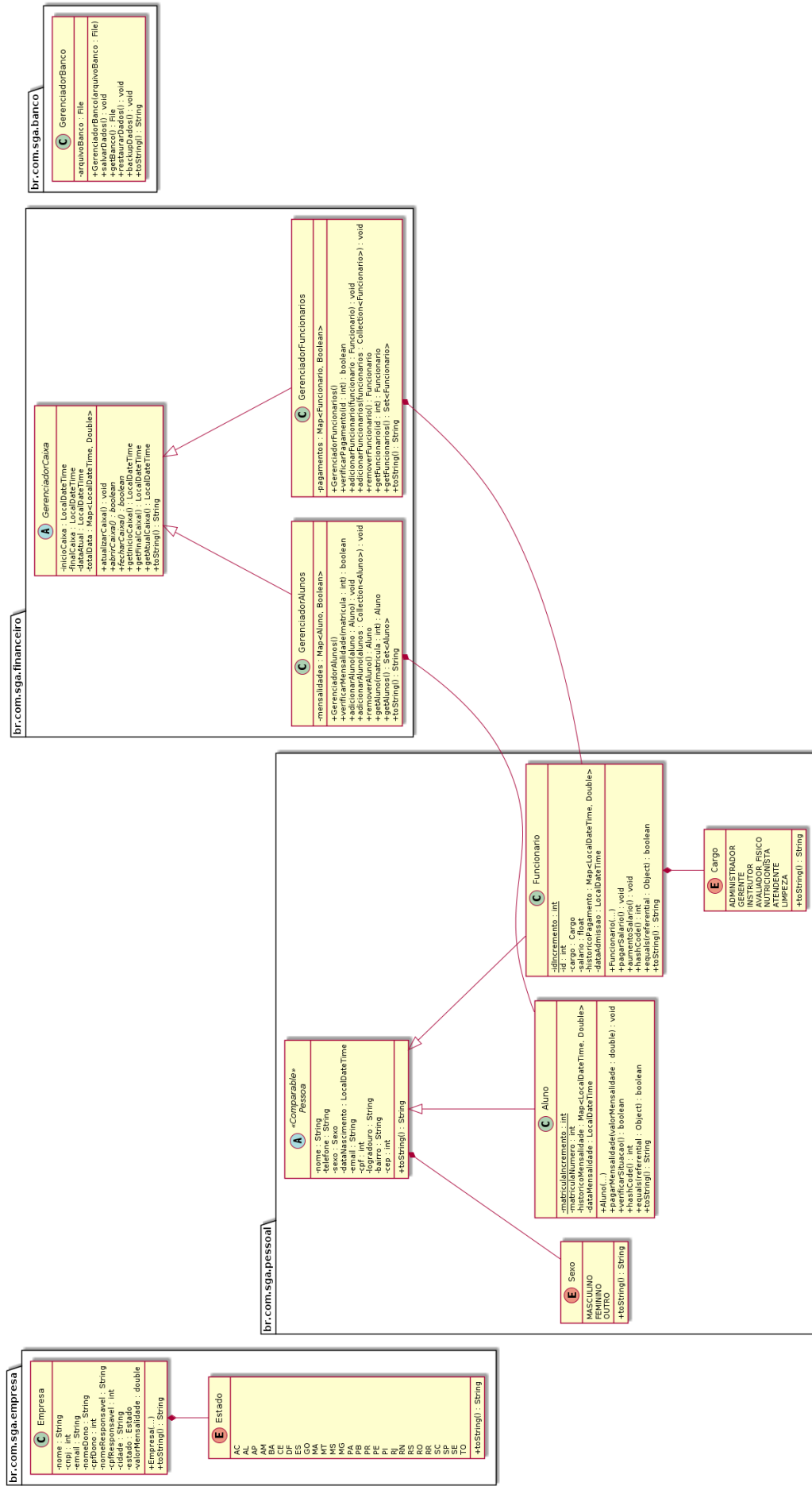
O projeto foi pensado partindo do "pessoal": alunos e funcionários. "Como representar da melhor maneira possível os objetos que movimentam a empresa?". Depois, foi pensado na parte financeira, isto é, como organizar o caixa da empresa. Dividi-lo em duas partes, onde cada uma irá gerenciar um dos objetos centrais foi a solução. Ao final foi pensando em como modelar a empresa que usará o sistema, bem como o gerenciador de banco de dados.

O projeto está estruturado em pacotes, utilizando a notação "br.com.sga.pacote", para assegurar a sua organização. Foram utilizadas classes abstratas, para assegurar polimorfismo e reuso de código, assim como enuns para a organização.

Foram utilizados vários mapeamentos nas classes, para assegurar rapidez e organização. A classe Aluno mapeia os valores pagos de mensalidade a uma data. Já a classe funcionário mapeia os valores pagos de salário a uma data. Caso nulo o pagamento não foi realizado. O mesmo vale para as classes gerenciadoras, que mapeiam um funcionário, ou aluno, para a sua situação quanto ao mês corrente. Essas classes de gerência se baseiam na classe "GerenciadorCaixa", que irá prover os atributos e métodos para controlar o caixa. A classe Pessoa implementa a interface Comparable, dessa forma exigindo que seus descendentes implementem os métodos requeridos.

Segue, na próxima página, o diagrama UML do projeto. Vale dizer que, por ter uma grande quantidade de atributos, a representação dos métodos getters e setters, assim como os construtores, teve que ser reduzida ao mínimo para caber no diagrama.

4.1 Diagrama UML



5 Conclusão

O repositório do projeto pode ser encontrado no [GitHub](#), tendo toda a estrutura do projeto, bem como o diagrama UML.