



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

PROGRAMAÇÃO ORIENTADA A OBJETOS

*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 |
| 5 | Funcionalidades | 3 |
| 5.1 | Primeiro acesso e Definição dos dados da empresa | 4 |
| 5.2 | Gerenciamento da Empresa | 4 |
| 5.3 | Gerenciamento de Usuários | 4 |
| 5.4 | Gerenciamento de alunos | 5 |
| 5.4.1 | Adicionar um aluno | 5 |
| 5.4.2 | Listar os alunos | 5 |
| 5.4.3 | Buscar aluno por matrícula | 6 |
| 5.4.4 | Atualizar dados do aluno | 6 |
| 5.4.5 | Remover um aluno | 6 |
| 5.4.6 | Listagem dos pagamentos de alunos | 6 |
| 5.5 | Gerenciamento de Pagamentos | 6 |
| 5.5.1 | Abrir e Fechar o Caixa do dia | 6 |
| 5.5.2 | Fazer pagamento de mensalidade do aluno | 7 |
| 5.5.3 | Listar alunos baseado na mensalidade | 7 |
| 5.5.4 | Listar histórico do caixa | 7 |
| 5.6 | Controle de Acesso | 7 |
| 5.7 | Configurações do Sistema | 8 |
| 6 | Diagrama UML | 9 |

1 Introdução

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

O projeto partiu de uma proposta mais generalizada e fluíu em uma análise mais específica, dando origem a um sistema sólido e robusto. A ideia de construir um sistema simples, porém completa, que atenda as principais necessidades de uma pequena academia, como administrar o cadastro de alunos e de organizar o pagamento de mensalidades.

Partindo de uma visão menos simplista um sistema como esse tem alguns OUTROS pontos em que um gerenciamento mais efetivo deve ser usado, como organização de usuários que tem acesso ao sistema e um sistema de banco de dados simples, porém organizado.

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

Repositório: <https://github.com/pedrobotelho15/SGA>

Ferramentas Utilizadas: Java 16, PlantUML, Discord, Visual Studio Code.

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 de alunos e usuários, bem como do controle do caixa, 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. A interface foi cuidadosamente projetada para funcionar da melhor maneira possível, sem encher a tela do usuário, esperando pelo *input* do usuário para continuar e usando códigos de escape ANSI para escrever palavras com cores.

O sistema foi encapsulado em um executável JAR, de forma que fique mais fácil a inicialização do sistema pelo cliente, sendo necessário apenas manejar um único lançador, e, como não será utilizado um sistema de banco de dados com servidor local, como MySQL, também não será necessário realizar mais passos, pois o banco de dados em arquivo fica salvo dentro do JAR, ou seja, não é necessário transportar pastas, apenas o arquivo JAR!

Os alunos, bem como os usuários, serão gerenciados por classes de gerenciamento próprias, que irão manejar as coleções de dados necessárias, como listas de usuários, alunos, mensalidades, etc... Temos então classes que irão modelar algo dentro do sistema, e classes que irão agir como utilitários, mesmo que sejam instanciados, para controlar o banco de dados.

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 tem 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:

- Cadastro das informações da empresa
- Cadastro e gerenciamento de Alunos e Controle de Acesso
- Cadastro e gerenciamento de Usuários
- Sistema de Mensalidade e Gerenciamento de Caixa
- Gerenciamento de Banco de Dados local e Backup
- Interface de Linha de Comando Turbinada

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 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 enums para a organização.

O projeto foi pensado partindo do "pessoal": alunos e funcionários. "Como representar da melhor maneira possível os objetos que movimentam a empresa?". Acabamos por utilizar usuários do sistema ao invés de funcionários. Depois, foi pensado na parte financeira, isto é, como organizar o caixa da empresa. Ao final foi pensando em como modelar a empresa que usará o sistema, bem como o gerenciador de banco de dados.

Foram utilizados várias coleções de dados nas classes, para assegurar rapidez e organização. A classe Aluno lista todos os pagamentos, por exemplo. Já a classe Caixa mapeia uma data aos valores pagos de mensalidade. A classe Pessoa implementa a interface Comparable, dessa forma exigindo que seus descendentes implementem os métodos requeridos.

Para simplicidade o banco de dados foi feito a partir de serialização de objetos. Sendo assim, todas as classes que são salvas no banco de dados pelas classes gerenciadores são "serializáveis"

5 Funcionalidades

O sistema tem diversas funcionalidades, as quais são listadas abaixo. Vale lembrar que o sistema é encapsulado em um JAR e tem uma interface otimizada para o cliente, o que facilita sua utilização.

O sistema conta com gerenciamento avançado do banco de dados, com controle de backup, e muitas outras funcionalidades, tudo encapsulado em um JAR! Isso facilita a utilização do usuário. O menu também é muito fácil de utilizar:

```
-----
Sistema de Gerenciamento de Academias (SGA)
-----

1) Controle de Entrada
2) Gerenciar Empresa
3) Gerenciar Usuários
4) Gerenciar Alunos
5) Gerenciar Pagamentos
6) Configurações do Sistema
7) Sair do Sistema
> █
```

5.1 Primeiro acesso e Definição dos dados da empresa

Ao inicializar o sistema pela primeira vez não haverá nada no banco de dados. Dessa forma, será necessário configurar as informações da empresa e definir um usuário administrador do sistema. O sistema irá pedir para o usuário configurar a empresa, indicando erros e pedindo para inserir novamente os dados caso sejam inválidos. O sistema realiza uma verificação toda vez que um dado de CPF, CNPJ, CEP ou telefone é inserido.

Após criar a empresa o sistema irá pedir para criar um usuário. Logo, o usuário do sistema irá definir um login e uma senha para acessá-lo. Após isso irá informá-los para fazer login. Assim terá acesso ao sistema por inteiro! São 4 menus de gerência, além do menu principal.

5.2 Gerenciamento da Empresa

É possível modificar as informações definidas da empresa, no menu de gerenciamento da empresa, assim como listar os dados no menu principal. Só é possível acessar o menu da empresa caso o usuário tenha permissões de administrador!

```
-----
Sistema de Gerenciamento de Academias (SGA)
-----
1) Mostrar Dados da Empresa
2) Modificar o Nome da Empresa
3) Mudar o CNPJ da Empresa
4) Trocar o e-mail da Empresa
5) Modificar o endereço da Empresa
6) Retornar
> █
```

5.3 Gerenciamento de Usuários

No menu de usuários é possível realizar mudanças nos usuários cadastrados, inclusive no usuário criado no começo do programa. Além disso é possível listar os usuários cadastrados, adicionar usuário, indicando seu login, senha, e se tem permissões de administrador, atualizar informações do usuário e remover um usuário baseado em seu ID. Uma boa prática foi mapear os IDs dos usuários aos usuários na classe de gerência, facilitando a busca.

A tela dos usuários fornece, então, algumas opções de CRUD, apenas o necessário para o bom funcionamento do sistema.

```
-----
Sistema de Gerenciamento de Academias (SGA)
-----

1) Adicionar um Usuário
2) Listar os Usuários
3) Atualizar o Login de um Usuário
4) Atualizar a Senha de um Usuário
5) Atualizar a Permissão de um Usuário
6) Remover um Usuário
7) Retornar
> 
```

5.4 Gerenciamento de alunos

O gerenciamento de alunos se dá de uma maneira um pouco diferente. Com muitos atributos a atualização de dados é feita em uma só opção, e é possível escolher se um atributo será modificado ou não. Além do clássico CRUD ainda temos a funcionalidade de listar as mensalidades pagas por um aluno, informada sua matrícula.

A tela dos clientes é bem mais robusta e completa, tendo como realizar uma atualização mais eficiente. A parte da leitura de dados é bem mais eficiente e diversificada também.

```
-----
Sistema de Gerenciamento de Academias (SGA)
-----

1) Adicionar um Aluno
2) Listar os Alunos
3) Buscar Aluno por matrícula
4) Listar mensalidades pagas por aluno
5) Atualizar um Aluno
6) Remover um Aluno
7) Retornar
> 
```

5.4.1 Adicionar um aluno

Funcionalidade na qual é permitido ao usuário logado fazer o cadastro de um aluno no banco de dados da academia. São requisitados vários dados para o cadastro, como nome, telefone, sexo, e-mail, CPF, data de nascimento, assim como também dados do seu endereço.

5.4.2 Listar os alunos

Funcionalidade na qual são listados os alunos com seu nome e matrícula

5.4.3 Buscar aluno por matrícula

Funcionalidade na qual é buscado um aluno através da sua matrícula. São listados todos os dados informados no momento do cadastro.

5.4.4 Atualizar dados do aluno

Funcionalidade na qual é permitida a atualização das informações de um aluno.

5.4.5 Remover um aluno

Funcionalidade na qual é possível remover um aluno informando apenas sua matrícula.

5.4.6 Listagem dos pagamentos de alunos

Listar todos os pagamentos de um aluno, informando apenas sua matrícula.

5.5 Gerenciamento de Pagamentos

O menu do gerenciamento de pagamentos inclui um gerenciador de caixa muito semelhante ao usado em comércios: abre-se o caixa, realiza o pagamento de mensalidades, e o fecha ao final do dia. O histórico de caixas ficará salvo no sistema, bem como o pagamento de cada mensalidade do aluno. É possível listar os alunos com mensalidades atrasadas e em dia.

```
-----
Sistema de Gerenciamento de Academias (SGA)
-----

1) Abrir caixa do dia
2) Fechar caixa do dia
3) Fazer pagamento da mensalidade do Aluno
4) Listar alunos com mensalidade em dia
5) Listar alunos com mensalidade atrasada
6) Listar o histórico do caixa
7) Retornar
> 
```

5.5.1 Abrir e Fechar o Caixa do dia

É possível abrir o caixa, permitindo que sejam feitos pagamentos de mensalidades de alunos, assim como fechá-lo, totalizando o saldo do dia.

5.5.2 Fazer pagamento de mensalidade do aluno

Funcionalidade na qual é permitido fazer o pagamento da mensalidade do aluno, informando apenas a matrícula e o valor da mensalidade.

5.5.3 Listar alunos baseado na mensalidade

É possível listar os alunos os quais estão com a mensalidade em dia ou atrasada, mostrando seu nome e sua matrícula.

5.5.4 Listar histórico do caixa

Funcionalidade no qual é informado todo o histórico do caixa.

5.6 Controle de Acesso

É possível realizar o controle de acesso de alunos, ou seja, determinar quem pode entrar ou não no estabelecimento baseado na sua situação cadastral, isso é, se está com a mensalidade em dias. Esse controle é feito pelo menu “Controle de Acesso”. Deve-se inserir a matrícula do aluno. Então o sistema irá mostrar se a sua entrada é permitida ou não.

Se o usuário estiver com a mensalidade em dias será mostrada uma mensagem, em cor verde, “ACESSO PERMITIDO!!!”. Caso não, será mostrado “ACESSO NEGADO!!!”, em cor vermelha, mostrando o nome e CPF do aluno.

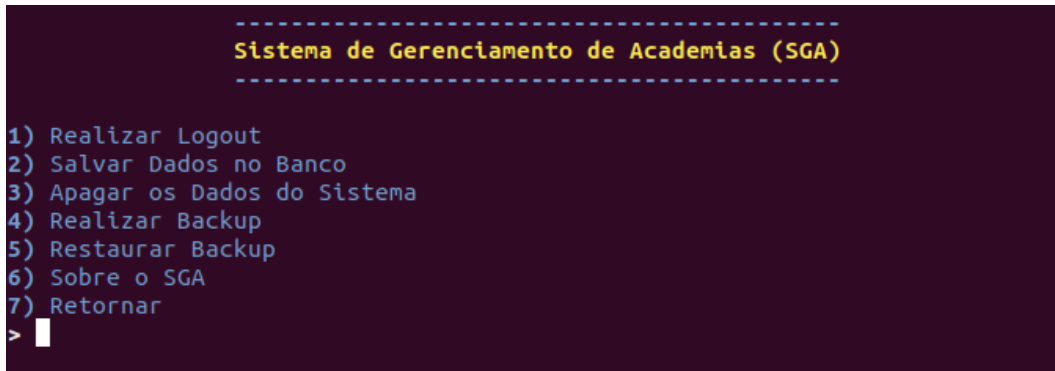
```
-----  
Sistema de Gerenciamento de Academias (SGA)  
-----  
  
Aluno: Atilio Luiz  
CPF: 11886936382  
  
ACESSO PERMITIDO!!!  
  
Pressione ENTER para voltar ao controle de acesso.  
█
```

```
-----  
Sistema de Gerenciamento de Academias (SGA)  
-----  
  
Aluno: Wagner Guimaraes  
CPF: 44685279352  
  
ACESSO NEGADO!!!  
  
Pressione ENTER para voltar ao controle de acesso.  
█
```

5.7 Configurações do Sistema

O menu de configurações do sistema adiciona um novo sabor ao controle do banco de dados: sistema de backups. Agora é possível realizar e restaurar backups, além de reverter o sistema para os padrões de fábrica, apagando todos os dados.

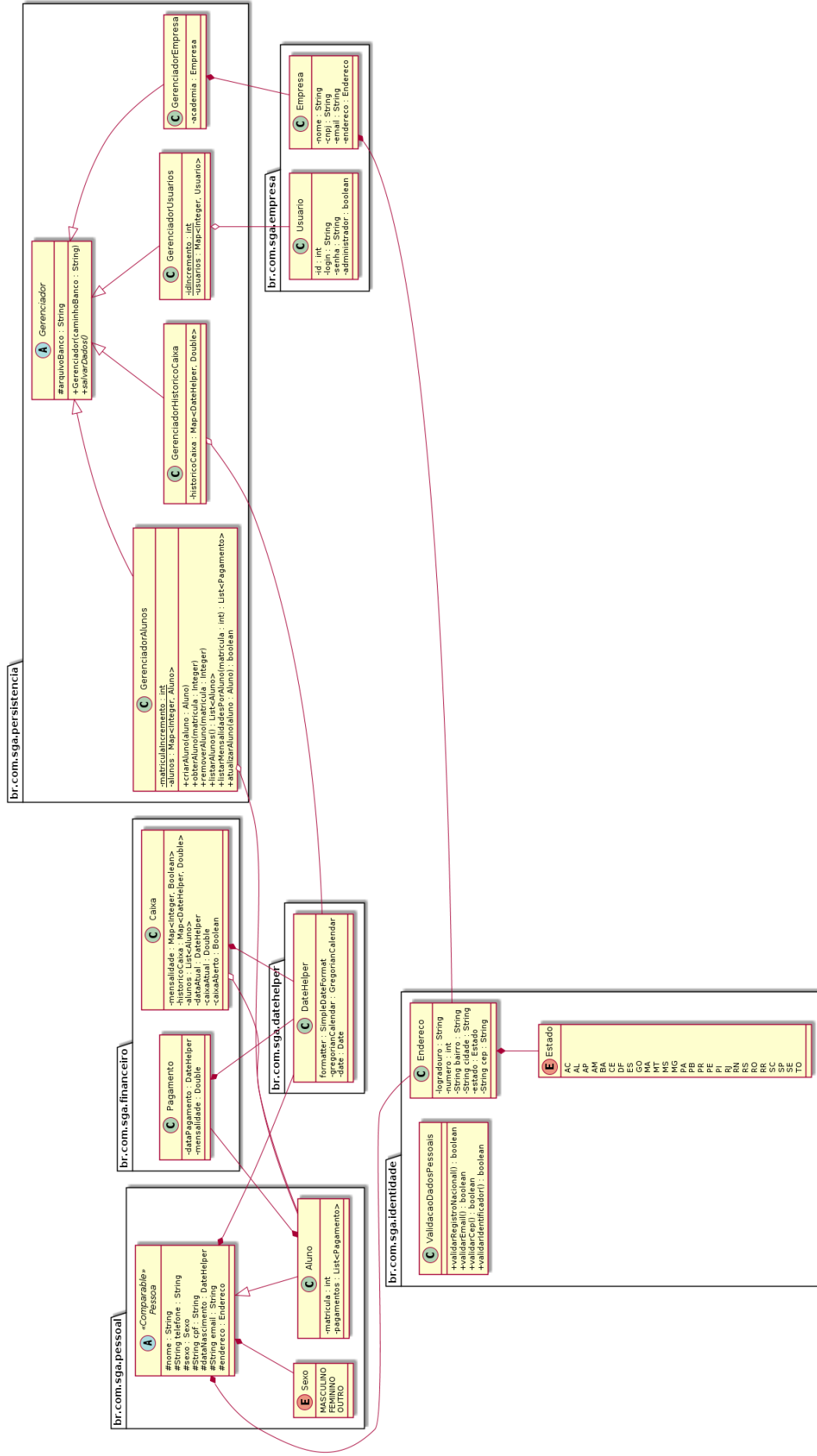
Também é possível fazer logout, tendo que fazer o login novamente, bem como salvar os dados correntes do sistema no banco, coisa que em outra ocasião só é feito na saída do sistema. Uma opção para ver a versão e outras informações sobre o sistema também foi adicionada.

A terminal window with a dark purple background. At the top, the text "Sistema de Gerenciamento de Academias (SGA)" is displayed in yellow, flanked by dashed blue lines. Below this, a list of seven options is shown in white text, each preceded by a number: 1) Realizar Logout, 2) Salvar Dados no Banco, 3) Apagar os Dados do Sistema, 4) Realizar Backup, 5) Restaurar Backup, 6) Sobre o SGA, and 7) Retornar. At the bottom of the list, there is a white prompt character ">" followed by a small white cursor block.

```
Sistema de Gerenciamento de Academias (SGA)
-----
1) Realizar Logout
2) Salvar Dados no Banco
3) Apagar os Dados do Sistema
4) Realizar Backup
5) Restaurar Backup
6) Sobre o SGA
7) Retornar
> █
```

O JAR foi gerado sem um banco de dados, mas para quem quiser testar alguns dados foram disponibilizados nos arquivos do diretório database/.

6 Diagrama UML



7 Conclusão

O sistema foi um grande desafio, no sentido de fazer o planejamento de um programa completo, e que atendesse as necessidades de uma academia, em um prazo tão curto! Mudamos o plano de ação muitas vezes até chegarmos ao resultado final.

Foi muito interessante e de muito aprendizado o desenvolvimento do programa, e com certeza fará parte de nosso portfólio agora, já que colocamos todos nossos esforços em fazer um projeto bem feito e satisfatório!

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