

UNIVERSIDADE ESTADUAL PAULISTA

“JÚLIO DE MESQUITA FILHO”

Instituto de Geociências e Ciências Exatas - IGCE

Curso de Bacharelado em Ciências da Computação

ADRIANO YOSHIMOTO

GABRIEL LUIZ

LEO EDUARDO

GUILHERME SIMIONATTO

**SISTEMA DE MANUTENÇÃO DE
MENSALIDADES DE ACADEMIA COM
DIVERSAS MODALIDADES**

Orientadora: Profa. Dra. Simone das Graças Domingues Prado

Rio Claro - SP

2019

SISTEMA DE MANUTENÇÃO DE MENSALIDADES DE ACADEMIA COM DIVERSAS MODALIDADES

Relatório de Linguagens Comerciais de Programação, realizado de janeiro à julho de 2019, visando desenvolver uma plataforma java pelo Curso de Bacharelado em Ciências da Computação do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de Rio Claro.

Adriano Yoshimoto
Alunos: Gabriel Luiz
Guilherme Simionato
Leo Eduardo

Orientadora: Profa. Dra. Simone das Graças Domingues Prado

Rio Claro - SP

2019

Sumário

1	INTRODUÇÃO	3
2	DESENVOLVIMENTO	4
2.1	Telas	4
2.1.1	Tela de Pesquisa	4
2.1.2	Tela de Cadastro	7
2.2	Back-End	9
3	TECNOLOGIAS	11
3.1	Tecnologias	11
3.1.1	Java 8 e Mavan	11
3.1.2	Spring e Lombok	11
3.1.3	MongoDB e ORM	11
3.1.4	IDE IntelliJ	11
3.1.5	GitHub	12
4	CONHECIMENTOS UTILIZADOS APRENDIDOS NA DISCIPLINA	13
4.1	Conhecimentos	13
5	CONCLUSÃO	14
	REFERÊNCIAS	15

1 Introdução

O objetivo deste projeto foi realizar um sistema de academia visando facilitar o processo de controle que diversas academias enfrentam devido aos diversos tipos de pagamentos possíveis neste caso o controle de pagamentos. Para isso, o usuário tem a possibilidade de efetuar cadastro dos alunos, modalidades, valores, data de pagamento entre outras informações. Uma vez que os dados estejam cadastrados, o usuário pode procurar a situação (paga/pendente) dos alunos e realizar o controle e baixa no sistema.

Sendo assim, o Capítulo 2 aborda o processo de desenvolvimento do software. Este, por sua vez, é subdividido: Capítulo 2.1 referente às telas e o Capítulo 2.2 referente às ferramentas usadas e particularidades do back-end.

Por fim, o Capítulo 5 destaca as considerações finais a respeito do projeto que engloba e utiliza os conceitos vistos durante as aulas de Linguagens Comerciais de Programação lecionadas no primeiro semestre de dois mil e dezenove.

2 Desenvolvimento

Usaremos as tecnologias de Java e Swing (código e telas), MongoDB (estruturar o banco de dados), Spring (comunicação com o banco de dados) e GIT (gerenciamento de versão) para criação deste sistema. Além disso, o atual escopo não prevê um controle financeiro (retorno de troco, quantidade recebida ou ainda fechamento de contas a pagar).

Outras informações pertinentes ao cadastro antes previstas como endereço, contato, desconto nas mensalidades entre outras também foram descartadas nesta versão e podem, futuramente, ser implementadas.

2.1 Telas

A aplicação conta com duas telas onde o usuário tem a liberdade de pesquisar e cadastrar novos usuários, conforme vemos a seguir.

2.1.1 Tela de Pesquisa

Assim que a aplicação é inicializada podemos observar (Figura 1) os filtros para a pesquisa de mensalidades dos alunos, um botão para pagamento e outro para efetuar os cadastros.

Gym Application

Cliente:

Modalidade:

Data Inicial: 27/06/2019

Data Final: 27/06/2019

Status: ☐ Pago

Pesquisar

ID	Cliente	Modali...	Data V...	Valor	Pago
----	---------	-----------	-----------	-------	------

ID:

Pagar

Cadastrar

Figura 1 – Tela de Pesquisa vazia
Fonte: pessoal.

No exemplo (Figura 2) é feita uma busca utilizando todos os filtros que retorna todas as mensalidades não pagas do cliente Léo vinculadas à modalidade musculação.

Gym Application

Cliente: Modalidade:

Data Inicial: Data Final:

Status: ☐ Pago

ID	Cliente	Modali...	Data V...	Valor	Pago
5d...	Leo	musc...	15/08...	100.0	Pende...
5d...	Leo	musc...	15/09...	100.0	Pende...
5d...	Leo	musc...	15/10...	100.0	Pende...
5d...	Leo	musc...	15/11...	100.0	Pende...
5d...	Leo	musc...	15/12...	100.0	Pende...

ID:

Figura 2 – Exemplo de pesquisa na Tela de Pesquisa
Fonte: pessoal.

Ainda é possível verificar na figura que houve o pagamento das parcelas referente aos meses 06 e 07.

Evidentemente, uma nova pesquisa com o filtro pago aplicado resulta na exibição das duas parcelas pagas (Figura 3).

Gym Application

Cliente: Modalidade:

Data Inicial: Data Final:

Status: ☒ Pago

ID	Cliente	Modali...	Data Ve...	Valor	Pago
5d...	Leo	musc...	15/06...	100.0	Pago
5d...	Leo	musc...	15/07...	100.0	Pago

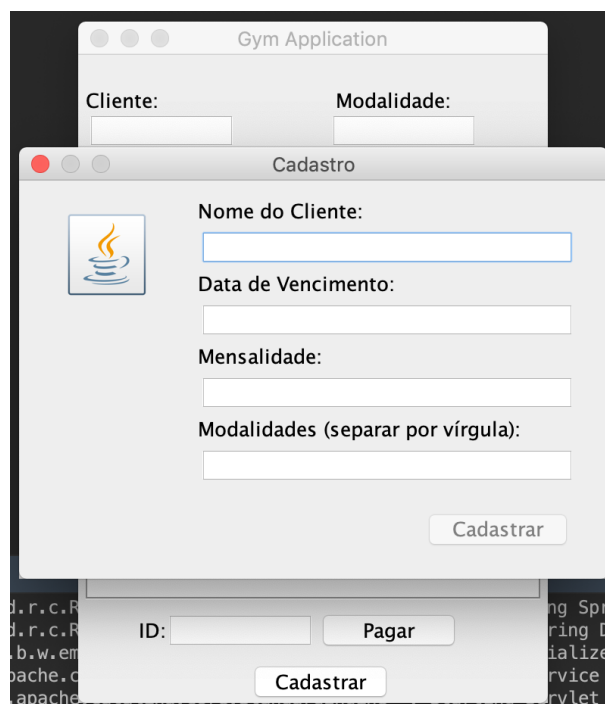
ID:

Figura 3 – Exemplo de pesquisa na Tela de Pesquisa

Fonte: pessoal.

2.1.2 Tela de Cadastro

A Tela de Pesquisa introduz o acesso ao botão Cadastrar que possibilita o usuário inserir novos alunos, vinculando as modalidades, valores e a data de vencimento da parcela (Figura 4).

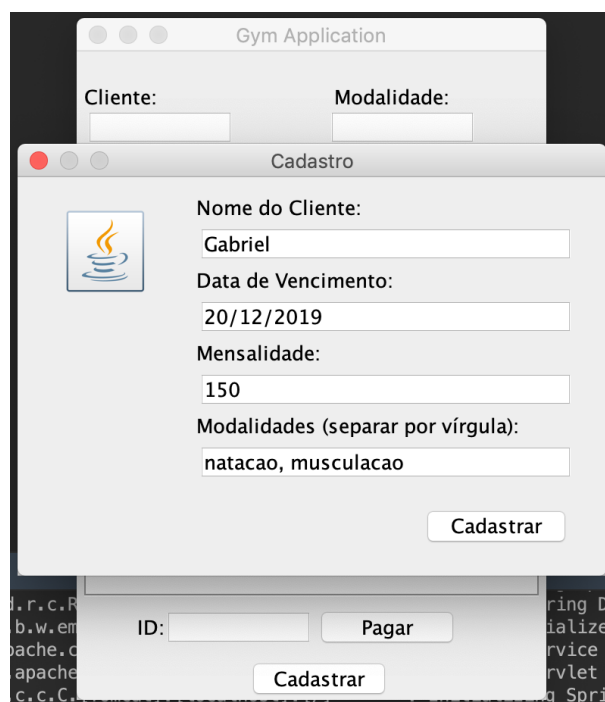


The screenshot shows a web application window titled "Gym Application". In the foreground, there is a modal dialog box titled "Cadastro". The dialog contains a Java logo on the left and several input fields on the right: "Nome do Cliente:", "Data de Vencimento:", "Mensalidade:", and "Modalidades (separar por vírgula):". A "Cadastrar" button is located at the bottom right of the dialog. In the background, another form is partially visible with fields for "Cliente:" and "Modalidade:", and buttons for "ID:", "Pagar", and "Cadastrar".

Figura 4 – Tela de Cadastro vazia
Fonte: pessoal.

Como podemos observar, os quatro campos devem ser preenchidos para que o cadastro seja efetuado.

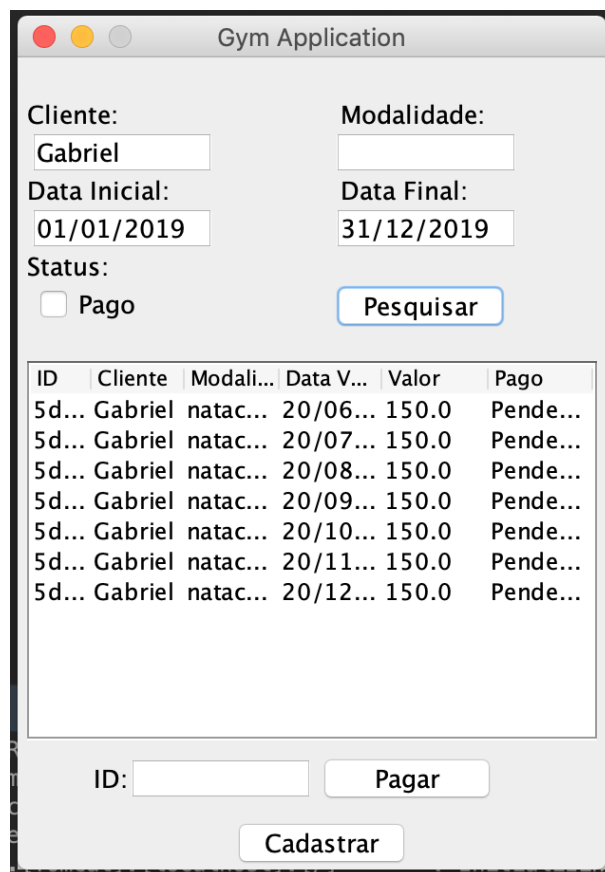
Uma vez que o preenchimento obrigatório ocorreu, é possível efetuar o cadastro (Figura 5) que poderá ser visualizado com os filtros adequados na Tela de Pesquisa.



This screenshot shows the same "Cadastro" dialog box as in Figure 4, but now it is filled with example data. The "Nome do Cliente:" field contains "Gabriel", "Data de Vencimento:" contains "20/12/2019", "Mensalidade:" contains "150", and "Modalidades (separar por vírgula):" contains "natacao, musculacao". The "Cadastrar" button remains at the bottom right. The background form is also visible, showing the same structure as in Figure 4.

Figura 5 – Exemplo de cadastro na Tela de Cadastro
Fonte: pessoal.

Evidentemente, a busca com os filtros adequados resultam no preenchimento da tabela com os dados cadastrados (Figura 6) e possibilita o usuário a realizar as baixas necessárias.



The screenshot shows a window titled "Gym Application". It contains search filters for "Cliente:" (Gabriel), "Modalidade:" (empty), "Data Inicial:" (01/01/2019), "Data Final:" (31/12/2019), and "Status:" (Pago checkbox). A "Pesquisar" button is present. Below the filters is a table with 6 columns: ID, Cliente, Modali..., Data V..., Valor, and Pago. The table displays 8 rows of data for Gabriel, all with a value of 150.0 and status "Pende...". At the bottom, there is an "ID:" field, a "Pagar" button, and a "Cadastrar" button.

ID	Cliente	Modali...	Data V...	Valor	Pago
5d...	Gabriel	natac...	20/06...	150.0	Pende...
5d...	Gabriel	natac...	20/07...	150.0	Pende...
5d...	Gabriel	natac...	20/08...	150.0	Pende...
5d...	Gabriel	natac...	20/09...	150.0	Pende...
5d...	Gabriel	natac...	20/10...	150.0	Pende...
5d...	Gabriel	natac...	20/11...	150.0	Pende...
5d...	Gabriel	natac...	20/12...	150.0	Pende...

Figura 6 – Exemplo de pesquisa na Tela de Cadastro
Fonte: pessoal.

2.2 Back-End

A construção das tabelas se dá conforme segue:

Cadastro			
Long idCadastro	String nomeCliente	String modalidades	Date diaVencimento

Figura 7 – Tabela cadastro
Fonte: pessoal.

Pagamento			
Long idPagamento	Long idCadastro	Long diaVencimento	Boolean pago

Figura 8 – Tabela Pagamento
Fonte: pessoal.

3 Tecnologias

3.1 Tecnologias

Neste trabalho foram usadas diversas tecnologias para auxiliar no processo de desenvolvimento, onde este projeto que poderia ser utilizado também para ser projetado telas web onde o backend foi desenvolvido para aceitar requisições web e salvar ou realizar pesquisas.

3.1.1 Java 8 e Mavan

Dentre estas tecnologias a linguagem Java na versão 8 (ou superior) conforme utilizada em sala de aula, maven como ferramenta de automação de compilação "O Maven utiliza um arquivo XML (POM) para descrever o projeto de software sendo construído, suas dependências sobre módulos e componentes externos."(WIKIPEDIA, 2019).

3.1.2 Spring e Lombok

É utilizado do spring para realizar a conexão com o banco de dados e expor os endpoints utilizados na aplicação, juntamente com o lombok que é utilizado para facilitar o desenvolvimento evitando ter que digitar todos os getters e setters e disponibilizando um builder para facilitar a maneira de instanciar um objeto.

3.1.3 MongoDB e ORM

Foi utilizado o banco de dados não relacional mongodb com ORM (Object Related Model), "ORM ou Object Relational Mapping é uma técnica de mapeamento objeto relacional que visa criar uma camada de mapeamento entre nosso modelo de objetos (aplicação) e nosso modelo relacional (banco de dados) de forma a abstrair o acesso ao mesmo."(PRISCILA, 20) onde as classes em java se tornam tabelas no banco de dados e é possível realizar consultas através da classe que estende o banco de dados.

3.1.4 IDE IntelliJ

Também foi utilizado da IDE intellij para facilitar o desenvolvimento "O IDE e um programa de computador, geralmente utilizado para aumentar a produtividade dos desenvolvedores de software, bem como a qualidade desses produtos. Podem auxiliar, através de ferramentas e características, na redução de erros e na aplicação de técnicas como o RAD (Rapid Application Development)"(SANTOS, 20?)

3.1.5 GitHub

Também foi feito o uso da ferramenta de controle de versão GitHub para gerenciar os arquivos de desenvolvimento e também deste relatório

4 Conhecimentos Utilizados aprendidos na disciplina

4.1 Conhecimentos

A disciplina de Linguagens Comerciais de Programação abordou diversos conceitos da linguagem java. Dentre eles conexão cliente e servidor, coleções, classes, tipos, swing para desenvolvimento das telas, threads, banco de dados,

5 Conclusão

Conforme exposto na seção anterior, utilizou-se dos conhecimentos lecionados pela professora Simone das Graças Domingues Prado durante o primeiro semestre de dois mil e dezenove para realização da aplicação que visa permitir a consulta de pagamentos pendentes e controle de matrícula dos alunos de uma academia. Além disso, para o desenvolvimento foi necessário o conhecimento e aprendizado de outras tecnologias não abordadas em aula, mas que complementam o conteúdo como Spring, MongoDB e GIT.

Os realizadores deste projeto acreditam ter aplicado os conhecimentos passados em aula e realizado um projeto pertinente ao proposto.

Também foi possível perceber que é totalmente viável um desenvolvimento de um software que possa atender a todos os tipos de academias

Referências

PRISCILA. *ORM - Object Relational Mapping - Revista Easy .Net Magazine* 28. 20. Disponível em: <<https://www.devmedia.com.br/orm-object-relational-mapping-revista-easy-net-magazine-28/27158>>. Citado na página 11.

SANTOS, A. K. dos. Os ide's (ambientes de desenvolvimento integrado) como ferramentas de trabalho em informatica. *Academico(a) do Curso de Ciência da Computação Universidade Federal de Santa Maria (UFSM)*, v. 1, n. 1, p. 1–1, 20? Citado na página 11.

WIKIPEDIA. *Apache Maven*. 2019. Disponível em: <https://pt.wikipedia.org/wiki/Apache_Maven>. Citado na página 11.