
Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Gabriel Fontana Junqueira Araújo

Pedro Francisco Isaac

Vinícius Jorge de Oliveira

Yuri Sabino do Prado

DESENVOLVIMENTO DE UM SISTEMA DE ACADEMIA EM JAVA FX

Gabriel Fontana Junqueira Araújo
Pedro Francisco Isaac
Vinícius Jorge de Oliveira
Yuri Sabino do Prado

DESENVOLVIMENTO DE UM SISTEMA DE ACADEMIA EM JAVAFX

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Faculdade de Tecnologia de Americana, sob a orientação metodológica do Prof. Me. Thiago Salhab Alves.

Área de concentração: Laboratório de Engenharia de *Software*.

Gabriel Fontana Junqueira Araújo

Pedro Francisco Isaac

Vinícius Jorge de Oliveira

Yuri Sabino do Prado

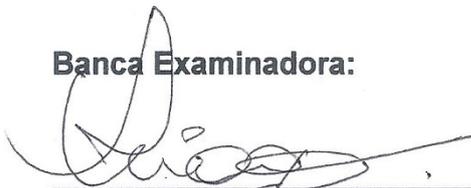
DESENVOLVIMENTO DE UM SISTEMA DE ACADEMIA EM JAVA FX

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Faculdade de Tecnologia de Americana, sob a orientação metodológica do Prof. Me. Thiago Salhab Alves.

Área de concentração: Laboratório de Engenharia de Software.

Americana, 1 de dezembro de 2022.

Banca Examinadora:



Thiago Salhab Alves

Mestre

Fatec de Americana



Rodrigo Brito Battilana

Mestre

Fatec de Americana



Lucas Serafim Perizotto

Especialista

Fatec de Americana

AGRADECIMENTOS

Aos nossos familiares, por todo o carinho, amor, incentivo e apoio emocional durante os momentos mais difíceis.

Ao orientador Me. Thiago Salhab Alves e demais professores do corpo docente, pela ajuda, paciência, amizade e principalmente pelas correções e ensinamentos.

À Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”, pelo ambiente de estudo agradável e recursos tecnológicos oferecidos.

À academia BoxGym, em especial ao proprietário, por sua confiança e tempo disponibilizado.

Aos nossos colegas de curso, com quem compartilhamos diversos momentos de aprendizado.

E a todos que participaram, seja direta ou indiretamente, da elaboração deste trabalho.

RESUMO

O presente trabalho acadêmico dedica-se fundamentalmente a documentar as etapas relacionadas à idealização e ao desenvolvimento de um sistema com versão *desktop* para auxiliar uma pequena academia de musculação e treinamento funcional denominada BoxGym, que está situada na cidade de Nova Odessa (SP). Este projeto inclui desde o levantamento de requisitos funcionais e não funcionais até a modelagem com diagramas e a apresentação e descrição das telas desenvolvidas. Dentre os recursos e ferramentas, o sistema fez uso principalmente da linguagem de programação Java e da plataforma JavaFX para a criação de uma aplicação moderna. Com a utilização dos conceitos descritos nas metodologias ágeis, especificamente na metodologia *Scrum*, foram definidos 11 ciclos de desenvolvimento, nos quais, ao final de cada um, pelo menos uma nova funcionalidade principal era adicionada ao sistema, mantendo um processo iterativo e incremental. Ao final, como resultado, a equipe obteve um sistema de academia funcional naquilo que propõe, com possibilidade para futuras adições e contribuições.

Palavras-chave: desenvolvimento de sistema; Java; JavaFX; metodologia *Scrum*.

ABSTRACT

The present academic work is fundamentally dedicated to documenting the steps related to the idealization and development of a desktop version system to help a small bodybuilding and functional training gym called BoxGym, which is located in the city of Nova Odessa (SP). This project includes from the survey of functional and non-functional requirements to the modeling with diagrams and the presentation and description of the screens developed. Among the resources and tools, the system used mainly the Java programming language and the JavaFX platform to create a modern application. Using the concepts described in agile methodologies, specifically in the Scrum methodology, 11 development cycles were defined, in which, at the end of each one, at least one new main functionality was added to the system, maintaining an iterative and incremental process. In the end, as a result, the team obtained a functional gym system in what it proposes, with the possibility for future additions and contributions.

Keywords: system development; Java; JavaFX; Scrum methodology.

LISTA DE FIGURAS

Figura 1 – Diagrama de casos de uso.....	31
Figura 2 – Diagrama de classes.....	47
Figura 3 – Diagrama de sequência "Realizar cadastro de usuário"	48
Figura 4 – Diagrama de sequência "Efetuar <i>login</i> ".....	49
Figura 5 – Diagrama de sequência "Visualizar <i>dashboard</i> ".....	49
Figura 6 – Diagrama de sequência "Manter fornecedores".....	50
Figura 7 – Diagrama de sequência "Manter produtos".....	51
Figura 8 – Diagrama de sequência "Manter entradas de estoque"	52
Figura 9 – Diagrama de sequência "Manter clientes".....	53
Figura 10 – Diagrama de sequência "Manter vendas"	54
Figura 11 – Diagrama de sequência "Visualizar produtos inseridos"	55
Figura 12 – Diagrama de sequência "Manter exercícios".....	56
Figura 13 – Diagrama de sequência "Manter treinos"	57
Figura 14 – Diagrama de sequência "Visualizar exercícios inseridos"	58
Figura 15 – Diagrama de sequência "Manter fichas"	59
Figura 16 – Diagrama de sequência "Visualizar treinos inseridos"	60
Figura 17 – Diagrama de sequência "Manter medidas"	61
Figura 18 – Diagrama de sequência "Manter mensalidades".....	62
Figura 19 – Diagrama de sequência "Manter cobranças"	63
Figura 20 – Diagrama de sequência "Receber cobranças"	63
Figura 21 – Diagrama de sequência "Manter pagamentos"	64
Figura 22 – Modelo conceitual do banco de dados.....	67
Figura 23 – Modelo lógico do banco de dados.....	68
Figura 24 – Processos da metodologia <i>Scrum</i>	83
Figura 25 – Tela de <i>login</i>	107
Figura 26 – Tela de cadastro de usuário.....	108
Figura 27 – Barra de menu.....	108
Figura 28 – Primeiro item da barra de menu	109
Figura 29 – Segundo item da barra de menu	109
Figura 30 – Terceiro item da barra de menu	109
Figura 31 – Quarto item da barra de menu	109
Figura 32 – Quinto item da barra de menu.....	110

Figura 33 – Sexto item da barra de menu	110
Figura 34 – Sétimo item da barra de menu	110
Figura 35 – Tela de gerenciamento de clientes.....	111
Figura 36 – Tela de cadastro de clientes	112
Figura 37 – Tela de atualização de clientes	112
Figura 38 – Tela de gerenciamento de fornecedores.....	113
Figura 39 – Tela de cadastro de fornecedores.....	114
Figura 40 – Tela de atualização de fornecedores	114
Figura 41 – Tela de gerenciamento de mensalidades.....	115
Figura 42 – Tela de cadastro de mensalidades.....	116
Figura 43 – Tela de atualização de mensalidades	116
Figura 44 – Tela de gerenciamento das mensalidades a receber.....	117
Figura 45 – Tela de gerenciamento das vendas a receber	117
Figura 46 – Tela de recebimento de mensalidades e vendas	118
Figura 47 – Tela de visualização de pagamentos	118
Figura 48 – Tela de gerenciamento de vendas	119
Figura 49 – Tela de visualização de itens de venda.....	120
Figura 50 – Tela de cadastro de vendas	120
Figura 51 – Tela de gerenciamento de produtos.....	121
Figura 52 – Tela de cadastro de produtos.....	122
Figura 53 – Tela de atualização de produtos	122
Figura 54 – Tela de gerenciamento de entradas de estoque	123
Figura 55 – Tela de visualização de itens de estoque.....	124
Figura 56 – Tela de cadastro de entradas de estoque	124
Figura 57 – Tela de gerenciamento de exercícios.....	125
Figura 58 – Tela de cadastro de exercícios	126
Figura 59 – Tela de atualização de exercícios	126
Figura 60 – Tela de gerenciamento de treinos	127
Figura 61 – Tela de visualização dos exercícios contidos nos treinos	128
Figura 62 – Tela de cadastro de treinos.....	128
Figura 63 – Tela de gerenciamento de fichas	129
Figura 64 – Tela de visualização dos treinos contidos nas fichas	130
Figura 65 – Tela de cadastro de fichas	130
Figura 66 – Tela de gerenciamento de medidas	131

Figura 67 – Tela de cadastro de medidas	132
Figura 68 – Tela de atualização de medidas.....	132
Figura 69 – <i>Dashboard</i> de clientes.....	133
Figura 70 – <i>Dashboard</i> de fornecedores	134
Figura 71 – <i>Dashboard</i> de produtos.....	134
Figura 72 – <i>Dashboard</i> de vendas	135

LISTA DE GRÁFICOS

Gráfico 1 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 1	85
Gráfico 2 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 2	88
Gráfico 3 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 3	91
Gráfico 4 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 4	93
Gráfico 5 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 5	95
Gráfico 6 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 6	97
Gráfico 7 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 7	99
Gráfico 8 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 8	101
Gráfico 9 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 9	103
Gráfico 10 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 10	104
Gráfico 11 – Gráfico de <i>Burndown</i> da <i>Sprint</i> 11	106

LISTA DE QUADROS

Quadro 1 – Comparativo de características e funcionalidades entre os sistemas	19
Quadro 2 – Requisitos funcionais do sistema	21
Quadro 3 – Requisitos não funcionais do sistema	22
Quadro 4 – Descrição do caso de uso "Realizar cadastro de usuário"	32
Quadro 5 – Descrição do caso de uso "Efetuar <i>login</i> "	32
Quadro 6 – Descrição do caso de uso "Manter fornecedores"	33
Quadro 7 – Descrição do caso de uso "Manter produtos"	34
Quadro 8 – Descrição do caso de uso "Manter entradas de estoque"	35
Quadro 9 – Descrição do caso de uso "Manter clientes"	36
Quadro 10 – Descrição do caso de uso "Manter vendas"	37
Quadro 11 – Descrição do caso de uso "Manter exercícios"	38
Quadro 12 – Descrição do caso de uso "Manter treinos"	39
Quadro 13 – Descrição do caso de uso "Manter fichas"	40
Quadro 14 – Descrição do caso de uso "Manter medidas"	41
Quadro 15 – Descrição do caso de uso "Manter mensalidades"	42
Quadro 16 – Descrição do caso de uso "Manter cobranças"	43
Quadro 17 – Descrição do caso de uso "Receber cobranças"	43
Quadro 18 – Descrição do caso de uso "Manter pagamentos"	44
Quadro 19 – Descrição do caso de uso "Visualizar produtos inseridos"	44
Quadro 20 – Descrição do caso de uso "Visualizar exercícios inseridos"	45
Quadro 21 – Descrição do caso de uso "Visualizar treinos inseridos"	45
Quadro 22 – Descrição do caso de uso "Visualizar <i>dashboard</i> "	45
Quadro 23 – Dicionário de dados da entidade <i>User</i>	69
Quadro 24 – Dicionário de dados da entidade <i>Supplier</i>	70
Quadro 25 – Dicionário de dados da entidade <i>Product</i>	71
Quadro 26 – Dicionário de dados da entidade <i>StockEntry</i>	71
Quadro 27 – Dicionário de dados da entidade <i>StockEntry_Product</i>	72
Quadro 28 – Dicionário de dados da entidade <i>Customer</i>	73
Quadro 29 – Dicionário de dados da entidade <i>Sale</i>	74
Quadro 30 – Dicionário de dados da entidade <i>Sale_Product</i>	74
Quadro 31 – Dicionário de dados da entidade <i>Exercise</i>	75
Quadro 32 – Dicionário de dados da entidade <i>Workout</i>	75

Quadro 33 – Dicionário de dados da entidade <i>Workout_Exercise</i>	76
Quadro 34 – Dicionário de dados da entidade <i>Measurement</i>	77
Quadro 35 – Dicionário de dados da entidade <i>Sheet</i>	78
Quadro 36 – Dicionário de dados da entidade <i>Sheet_Workout</i>	78
Quadro 37 – Dicionário de dados da entidade <i>Membership</i>	79
Quadro 38 – Dicionário de dados da entidade <i>Billing</i>	79
Quadro 39 – Dicionário de dados da entidade <i>Payment</i>	80
Quadro 40 – Atividades realizadas durante a <i>Sprint 1</i>	85
Quadro 41 – Atividades realizadas durante a <i>Sprint 2</i>	88
Quadro 42 – Atividades realizadas durante a <i>Sprint 3</i>	90
Quadro 43 – Atividades realizadas durante a <i>Sprint 4</i>	92
Quadro 44 – Atividades realizadas durante a <i>Sprint 5</i>	94
Quadro 45 – Atividades realizadas durante a <i>Sprint 6</i>	96
Quadro 46 – Atividades realizadas durante a <i>Sprint 7</i>	98
Quadro 47 – Atividades realizadas durante a <i>Sprint 8</i>	100
Quadro 48 – Atividades realizadas durante a <i>Sprint 9</i>	102
Quadro 49 – Atividades realizadas durante a <i>Sprint 10</i>	104
Quadro 50 – Atividades realizadas durante a <i>Sprint 11</i>	105

SUMÁRIO

1	INTRODUÇÃO	15
2	REQUISITOS.....	18
2.1	Análise dos concorrentes	18
2.2	Levantamento de requisitos.....	19
2.2.1	Requisitos funcionais	21
2.2.2	Requisitos não funcionais	21
2.3	Recursos e ferramentas	22
2.3.1	AnyDesk	23
2.3.2	Apache NetBeans	23
2.3.3	Astah UML	24
2.3.4	brModelo	24
2.3.5	CSS	24
2.3.6	Git.....	25
2.3.7	GitHub.....	25
2.3.8	Java.....	26
2.3.9	JavaFX	26
2.3.10	Scene Builder	27
2.3.11	XAMPP.....	28
3	MODELAGEM	29
3.1	Diagrama de casos de uso	29
3.2	Diagrama de classes.....	46
3.3	Diagrama de sequência	48
3.4	Banco de dados.....	64
3.4.1	Diagrama entidade-relacionamento	66
3.4.2	Dicionário de dados	69
4	PROJETO.....	81
4.1	Metodologia	81
4.2	Etapas de desenvolvimento	84
4.2.1	<i>Sprint 1</i>	84
4.2.2	<i>Sprint 2</i>	87
4.2.3	<i>Sprint 3</i>	89
4.2.4	<i>Sprint 4</i>	91

4.2.5	<i>Sprint 5</i>	93
4.2.6	<i>Sprint 6</i>	95
4.2.7	<i>Sprint 7</i>	97
4.2.8	<i>Sprint 8</i>	99
4.2.9	<i>Sprint 9</i>	101
4.2.10	<i>Sprint 10</i>	103
4.2.11	<i>Sprint 11</i>	105
4.3	Interfaces	107
5	CONSIDERAÇÕES FINAIS	136
	REFERÊNCIAS.....	138

1 INTRODUÇÃO

A palavra “bem-estar”, como aponta o Global Wellness Institute (GWI) (2021, p. 1, tradução nossa), pode ser definida “como a busca ativa de atividades, escolhas e estilos de vida que levam a um estado de saúde holístico”. Logo, “o bem-estar não é um estado passivo ou estático, mas sim uma ‘busca ativa’ associada a intenções, escolhas e ações”. Por ser um conceito holístico, “incorpora muitas dimensões diferentes”, dentre elas, a dimensão da atividade física, que geralmente é praticada de maneira recreativa em um local especializado para atividades de condicionamento físico (*fitness*) como uma academia, que será o foco deste trabalho.

Academia esportiva, ginásio ou simplesmente academia corresponde a um estabelecimento dedicado ao ensino e à realização de exercícios físicos que podem acontecer por meio de diferentes modalidades, por exemplo, musculação, *spinning*, artes marciais, ginástica, treinamento funcional, entre outras. Além dos equipamentos específicos de cada modalidade, as academias também costumam disponibilizar produtos para venda, atuando assim como comércio varejista de produtos alimentícios em geral e/ou comércio varejista de artigos esportivos.

Conforme o relatório *The Global Wellness Economy: Looking Beyond COVID* de 2021, o mercado e a economia de bem-estar (que englobam os negócios voltados às atividades e estilos de vida saudáveis) cresceram muito nos últimos anos, e isso se deve principalmente pelo fato dos consumidores estarem

[...] cada vez mais conscientes de que todos os aspectos de suas vidas – onde vivem; como eles trabalham; o que eles comeram; e como eles se exercitam, socializam e viajam – impactam seus resultados de saúde, resiliência mental e sensação geral de bem-estar (GLOBAL WELLNESS INSTITUTE, 2021, p. 11, tradução nossa).

Conseqüentemente, por estar contido no mercado de bem-estar, o setor *fitness* também se fortaleceu. De acordo com a International Health, Racquet & Sportsclub Association (IHRSA) (2020, tradução nossa) – uma associação que representa a indústria global de saúde e *fitness* e seus membros – em seu relatório *The 2020 IHRSA Global Report* (que fornece uma visão geral e abrangente dos mercados de academias em todo o mundo), o Brasil ocupa o segundo lugar em número de academias com 29.525 instalações, permanecendo atrás somente dos Estados Unidos.

Diante desse aumento de procura por parte dos clientes, em alguns casos, os proprietários das academias passaram a enfrentar dificuldades para lidar com a enorme quantidade de dados necessários para a realização de uma boa gestão de seu negócio; em outros, o problema que já existia, porém em menor intensidade, se agravou.

Levando em consideração as informações supracitadas, optou-se por atender a BoxGym, uma pequena academia de musculação e treinamento funcional localizada em Nova Odessa (SP). O local enfrenta problemas relativos à gestão de clientes, produtos e pagamentos desde 2005, época em que ainda era conhecida pelo nome Músculo em Ação.

Considerando o contexto, por ser uma academia de baixo custo, o investimento em tecnologia nunca foi prioridade. Desde sua fundação, todo o gerenciamento vem sendo efetuado de modo manuscrito (com papel e caneta), ação que pode ser considerada um obstáculo oculto, atrapalhando na gestão da empresa. Para piorar a situação, a pandemia de COVID-19 em 2020 provocou o fechamento temporário do estabelecimento por três meses, além de uma perda considerável de clientes e uma consequente baixa no faturamento.

Atualmente, no período pós-pandemia, a empresa está se reestruturando com a volta e a chegada de novos clientes, que estão contribuindo para a normalização do faturamento. Com a finalidade de auxiliar nesse momento, idealizou-se um sistema de computador, o qual seria responsável por fornecer uma visão ampla da gestão da empresa.

Em vista disso, o presente trabalho tem por objetivo geral desenvolver um sistema *desktop* com foco em um estabelecimento do setor *fitness*. Para tal, os seguintes objetivos específicos devem ser alcançados: analisar os sistemas existentes no mercado que apresentam propostas semelhantes; aplicar os conhecimentos adquiridos relativos à engenharia de *software* sobre a questão do levantamento de requisitos; compreender os recursos e ferramentas essenciais para a construção do projeto; utilizar técnicas de análise e modelagem de sistemas; empregar os princípios descritos nas metodologias ágeis, em especial na metodologia *Scrum*.

A estruturação do trabalho foi feita em cinco capítulos principais. O atual capítulo fornece uma contextualização sobre as academias esportivas, define uma academia como foco de estudo e expõe as motivações para isso, apresenta o objetivo

geral e como a equipe pretende alcançá-lo por meio dos objetivos específicos e finaliza dessa maneira, com um breve resumo dos conteúdos que serão encontrados no decorrer do projeto. O segundo capítulo dedica-se à documentação das questões realizadas no início do projeto, que incluem o comparativo de sistemas relacionados ao tema, levantamento de requisitos e revisão bibliográfica dos recursos e ferramentas utilizados. O terceiro capítulo foca na fase da modelagem, ou seja, apresenta, com auxílio de diversos quadros e diagramas, as estruturas que irão compor o sistema. O quarto capítulo descreve a metodologia empregada, as etapas e os resultados do desenvolvimento. Por fim, no quinto capítulo, são apresentadas as considerações finais, que tratam, de modo geral, do desfecho dos objetivos propostos, das contribuições para a equipe e das sugestões para estudos futuros.

2 REQUISITOS

O presente capítulo é dedicado a documentar os aspectos iniciais do projeto, incluindo desde a análise dos concorrentes, por meio da pesquisa de características e funcionalidades comuns aos sistemas de academia, até o levantamento de requisitos funcionais e não funcionais e a revisão bibliográfica dos recursos e ferramentas utilizados durante o desenvolvimento.

2.1 Análise dos concorrentes

Tendo em vista que o objetivo do trabalho é o desenvolvimento de um sistema *desktop*, foi feita uma breve exploração, por meio de *sites*, de sistemas brasileiros que possuem a mesma finalidade, isto é, voltados para academias. Essa pesquisa ocorreu no início do projeto, em 14 de fevereiro de 2022, logo após a escolha do tema com o intuito de analisar as tendências desse mercado. Como resultado, a equipe encontrou três sistemas concorrentes: 4GYM Free, Sistema SCA e DM10Fit, que serão descritos a seguir:

- a) 4GYM Free: trata-se de um sistema gratuito e básico criado pela Mobile Mind para academias e estúdios que ainda não utilizam uma solução informatizada e querem experimentar algo novo;
- b) Sistema SCA: é um sistema multiplataforma criado pela ProSistemas com foco em academias, clubes e estúdios. Suas funcionalidades estão relacionadas aos principais setores da empresa como financeiro, administrativo e comercial. Além disso, oferece uma versão *online*, *mobile* e *desktop*, conseguindo atender assim uma gama maior de clientes. Por ser completo e simples de utilizar, ele atende desde gestores menos experientes, que procuram apenas manter bons resultados em seu negócio, até gestores mais experientes, que necessitam de níveis mais complexos e avançados de gestão;
- c) DM10Fit: é um sistema multiplataforma desenvolvido pela empresa DM10 que auxilia gestores, professores e alunos. Para os gestores, ele permite acompanhar o crescimento do negócio e conta com diversos módulos para oferecer a melhor experiência aos alunos e à academia como um todo. Para os professores, o DM10Fit facilita a criação dos

treinos por meio de *smartphones* e/ou *tablets*. E para os alunos, fornece opções para consultar treinos, pagamentos, atualizações da academia, etc.

Levando em consideração as informações obtidas, o Quadro 1 apresenta um resumo das principais diferenças do sistema pensado para a academia BoxGym em relação aos mencionados acima. O símbolo “✓” indica que o *software* possui aquela determinada característica ou funcionalidade (no caso deste projeto, que possuirá ao final do desenvolvimento); e o símbolo “x”, que não possui (no caso deste projeto, que não possuirá).

Quadro 1 – Comparativo de características e funcionalidades entre os sistemas

	4GYM Free	Sistema SCA	DM10Fit	BoxGym
Multiplataforma	x	✓	✓	x
Múltiplas interfaces (gestor, professor e aluno)	x	✓	✓	x
Gratuito	✓	x	x	✓
Fornecedores	x	✓	x	✓
Produtos	x	✓	✓	✓
Estoque	x	✓	x	✓
Vendas	x	✓	x	✓
Clientes ou alunos	✓	✓	✓	✓
Treinos	x	✓	✓	✓
Medidas	x	✓	✓	✓
Mensalidades	x	✓	✓	✓
Financeiro	✓	✓	✓	✓

Fonte: Elaboração própria (2022).

2.2 Levantamento de requisitos

Conforme descrito pelo Institute of Electrical and Electronics Engineers (IEEE) (1990, p. 62, tradução nossa), o vocábulo “requisito” pode ser entendido, em um sentido mais amplo, como “uma condição ou capacidade necessária para um usuário resolver um problema ou atingir um objetivo”, ou ainda, em um sentido mais específico e relacionado com o campo da computação (principalmente da engenharia de *software*), como “uma condição ou capacidade que deve ser atendida ou possuída por um sistema ou componente do sistema para satisfazer um contrato, padrão, especificação ou outros documentos formalmente impostos”.

De modo semelhante, Sommerville (2011, p. 57) afirma que “os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento”. Portanto, entende-se que os requisitos refletem as necessidades e expectativas de um cliente ou usuário em relação a um produto de *software*.

Com base nessas definições, é possível compreender uma importante etapa presente na engenharia de *software* conhecida como engenharia de requisitos. Na visão de Sommerville (2011), corresponde ao processo de descobrir os requisitos de um sistema, analisá-los, documentá-los e verificá-los. Pressman (2011, p. 127) contribui para a compreensão desse conceito ao declarar que se trata do “amplo espectro de tarefas e técnicas que levam a um entendimento dos requisitos”, e menciona que ele abrange sete tarefas distintas, sendo uma delas o levantamento de requisitos. Sobre essa tarefa específica, Pressman (2011, p. 128) comenta:

Certamente parece bastante simples – pergunte ao cliente, aos usuários e aos demais interessados quais são os objetivos para o sistema ou produto, o que deve ser alcançado, como o sistema ou produto atende às necessidades da empresa e, por fim, como o sistema ou produto deve ser utilizado no dia a dia. Mas isso não é simples – na verdade, é muito difícil.

Levando esse comentário em consideração, é possível justificar essa afirmativa – de que não se trata de uma etapa simples de concluir na prática – já que projetar e desenvolver um sistema que resolve o problema errado não atende as necessidades de ninguém, por essa razão é importante obter e especificar requisitos inteligíveis e consistentes.

Para evitar esse problema, Sommerville (2011, p. 58) propõe uma divisão para os requisitos conforme os diferentes níveis de descrição apresentados em cada um, criando assim dois termos: requisitos de usuário e requisitos de sistema. Requisitos de usuário “são declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar”. Por outro lado, requisitos de sistema “são descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de *software*. O documento de requisitos do sistema [...] deve definir exatamente o que deve ser implementado”. Além dessa divisão, o autor ainda discorre sobre outra maneira de classificar os requisitos de um sistema, a qual será tratada a seguir.

2.2.1 Requisitos funcionais

Na engenharia de *software*, os requisitos funcionais (RF) descrevem o que um sistema deve fazer ou fornecer. Representam as funcionalidades que precisam ser necessariamente incorporadas ao sistema pelos desenvolvedores como parte do contrato para permitir que os usuários finais atinjam seus objetivos. Os requisitos funcionais também podem ser compreendidos em termos de entradas e saídas, já que definem declarações de como o sistema deve reagir a entradas específicas, isto é, seu comportamento básico, e o que será produzido ou retornado ao final de cada função executada. Quando expressos na forma de requisitos de usuário, os requisitos funcionais são geralmente abstratos; requisitos de sistema funcionais, no entanto, descrevem em detalhes as funções do sistema (SOMMERVILLE, 2011). O Quadro 2 apresenta os requisitos de usuário funcionais deste projeto, obtidos com a técnica da entrevista informal.

Quadro 2 – Requisitos funcionais do sistema

Identificação	Requisito funcional	Prioridade
RF001	Realizar cadastro de usuário	Desejável
RF002	Efetuar <i>login</i>	Desejável
RF003	Visualizar <i>dashboard</i>	Importante
RF004	Manter fornecedores	Essencial
RF005	Manter produtos	Essencial
RF006	Manter entradas de estoque	Desejável
RF007	Manter clientes	Essencial
RF008	Manter vendas	Essencial
RF009	Manter exercícios	Desejável
RF010	Manter treinos	Essencial
RF011	Manter fichas	Desejável
RF012	Manter medidas	Essencial
RF013	Manter mensalidades	Essencial
RF014	Manter cobranças	Importante
RF015	Manter pagamentos	Importante

Fonte: Elaboração própria (2022).

2.2.2 Requisitos não funcionais

Sommerville (2011) explica que os requisitos não funcionais (RNF) não têm relação direta com as funcionalidades oferecidas pelo sistema a seus usuários. Nesse

caso, enquanto os requisitos funcionais definem o que deve ser feito a nível de componente, os requisitos não funcionais definem como o sistema deve fazê-lo (ou ser) e são normalmente mais críticos que os funcionais, pois são capazes de afetar a arquitetura do sistema como um todo. Além disso, podem ser provenientes de três fontes:

- a) requisitos de produto, que restringem o comportamento do sistema, por exemplo, em termos de desempenho, confiabilidade, segurança e usabilidade;
- b) requisitos organizacionais, derivados de políticas e procedimentos da organização do cliente e do desenvolvedor;
- c) e requisitos externos, que compreendem os requisitos reguladores, legais e éticos.

O Quadro 3 apresenta os requisitos não funcionais deste projeto, que foram adquiridos utilizando a técnica da entrevista informal.

Quadro 3 – Requisitos não funcionais do sistema

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	O sistema deve possuir uma interface gráfica do usuário simples e intuitiva.	Usabilidade	Essencial
RNF002	O usuário deve ser capaz de compreender todas as funções do sistema após duas horas de treinamento.	Usabilidade	Desejável
RNF003	O sistema deve manipular os dados de um banco de dados local.	<i>Software</i>	Desejável
RNF004	O sistema deve funcionar sem estar conectado à Internet.	Disponibilidade	Desejável
RNF005	O sistema deve ser capaz de ser mantido de forma rápida e eficaz ao longo de sua vida útil.	Manutenibilidade	Importante
RNF006	O sistema deve ser capaz de receber novas funcionalidades não previstas sem prejudicar as já existentes.	Extensibilidade	Essencial
RNF007	O sistema deve ser executado em um computador com processador Intel Core i3 de 2ª geração e 4GB de memória RAM.	<i>Hardware</i>	Essencial

Fonte: Elaboração própria (2022).

2.3 Recursos e ferramentas

Em seguida, serão listados os recursos e ferramentas – programas e linguagens computacionais – envolvidos nas etapas de planejamento e desenvolvimento deste

projeto, além de suas respectivas descrições. Quando necessário, a julgar por sua importância, a descrição do recurso ou ferramenta será mais detalhada, expondo, por exemplo, vantagens oferecidas ou outras informações relevantes.

2.3.1 AnyDesk

AnyDesk é um *software* proprietário multiplataforma de acesso remoto desenvolvido pela empresa AnyDesk Software GmbH. Com ele, é possível controlar um computador ou outro dispositivo que esteja executando o AnyDesk de qualquer lugar do mundo. Além disso, o aplicativo oferece acesso não supervisionado, transferência de arquivos, compartilhamento de tela, recursos de VPN (*Virtual Private Network*), entre outras funcionalidades (ANYDESK, 2017, tradução nossa).

2.3.2 Apache NetBeans

O Apache NetBeans é um ambiente de desenvolvimento integrado (IDE, do inglês *Integrated Development Environment*) de código aberto e gratuito usado na construção de aplicativos para plataformas *web*, *mobile* ou *desktop*. Atualmente mantido pela Apache Software Foundation (ASF) e Oracle Corporation, está disponível para ser instalado nos sistemas operacionais Windows, Mac, Linux e Solaris, e possui suporte a diversas linguagens de programação, como Java, C, C++, PHP e JavaScript (ORACLE, 2018).

Wielenga (2015, tradução nossa) explica que, de maneira semelhante ao funcionamento de outras IDEs, o Apache NetBeans é constituído por um conjunto de editores e ferramentas gráficas para a codificação eficiente e detecção rápida de problemas, as quais irão orientar o usuário desde o momento da criação do projeto, passando pela edição, até a compilação, depuração e empacotamento. Além disso, o Apache NetBeans dispõe de diversas funções que simplificam o processo de desenvolvimento, a saber: notificar e destacar problemas no código, exibir dicas e sugestões de preenchimento automático, documentação de classes e métodos, integração com ferramentas de versionamento como o Git, entre outras.

Para o desenvolvimento do sistema *desktop* em JavaFX, a IDE foi utilizada em sua versão 13 (lançada em 4 de março de 2022).

2.3.3 Astah UML

Na visão de Guedes (2011, p. 41), ferramentas CASE (*Computer-Aided Software Engineering*), “são *softwares* que, de alguma maneira, colaboram para a execução de uma ou mais atividades realizadas durante o processo de engenharia de *software*”. Com isso, dentre as diversas ferramentas CASE que suportam a linguagem UML (*Unified Modeling Language*) disponíveis no mercado, optou-se por recorrer ao Astah UML em sua versão 8.5.0, de 10 de março de 2022.

Desenvolvido pela empresa japonesa Change Vision e disponibilizado nas plataformas Windows, Mac e em algumas distribuições Linux, é uma ferramenta projetada especificamente para UML, permitindo, desse modo, a concepção de diagramas de classes, casos de uso, sequência, entre outros (ASTAH, 2022).

2.3.4 brModelo

Como descrito por Sis4.com (2020), o brModelo é um *software* gratuito e de código aberto dedicado ao ensino de modelagem de bancos de dados relacionais. Foi desenvolvido por brasileiros na linguagem Java tendo em vista a metodologia defendida por Carlos Alberto Heuser na obra *Projeto de banco de dados*.

2.3.5 CSS

Cascading Style Sheets (CSS), conforme Robbins (2018, tradução nossa), é uma linguagem tradicionalmente aplicada para descrever como os elementos de um documento são formatados e apresentados ao usuário, isto é, como irão se portar no sistema.

O conceito por trás de seu funcionamento é simples: cria-se um documento de folha de estilo que especifica as fontes, cores, espaçamento e outras características que proporcionam uma aparência exclusiva aos componentes de uma página. Em seguida, vinculam-se todas as páginas que devem ter essa aparência à folha de estilo (MELONI, 2011, tradução nossa).

No caso deste sistema, uma versão do CSS adaptada aos recursos do JavaFX (conhecida como JavaFX CSS) foi adotada. O motivo de sua origem é permitir que pessoas já familiarizadas com CSS para *web* possam usá-lo para personalizar e

aplicar temas em controles JavaFX e objetos gráficos de maneira natural. Isso permite, portanto, a mistura de estilos CSS para JavaFX e outros propósitos em uma única folha de estilos. Com a intenção de diferenciar as duas versões, as propriedades do JavaFX CSS adotam o prefixo “-fx-”, por exemplo, “-fx-background-color” (propriedade utilizada para preencher o plano de fundo de um determinado controle) (ORACLE, 2014a).

2.3.6 Git

Uma ferramenta capaz de rastrear diferentes versões de um *software* ou qualquer outro conteúdo é chamada genericamente de sistema de controle de versão, gerenciador de código-fonte, sistema de controle de revisão ou qualquer outra permutação com palavras semelhantes. Apesar de nomenclaturas diferentes, todas estão se referindo à mesma funcionalidade: criar e manter um repositório de conteúdo que registre todas as alterações, mantendo-as em um histórico completo (LOELIGER; MCCULLOUGH, 2012, tradução nossa).

Nesse sentido, criado por Linus Torvalds para apoiar o desenvolvimento do *kernel* Linux, Git é um sistema gratuito e de código aberto para rastrear alterações feitas em um conjunto de arquivos ao longo do tempo (SILVERMAN, 2013, tradução nossa). Embora seja mais utilizado por programadores para coordenar mudanças do código-fonte de um *software* em desenvolvimento, a ferramenta também é capaz de lidar, como alega o próprio Git (2022), com qualquer tipo de conteúdo ou porte (desde projetos pequenos até os grandes), sempre com velocidade e eficiência, além de ser fácil de usar e aprender.

2.3.7 GitHub

O projeto fez uso do GitHub, uma plataforma *online* da Microsoft para hospedagem de repositórios Git que permite que programadores cadastrados possam, por exemplo, se conectar, contribuir com diversos projetos, divulgar trabalhos e códigos, enviar e baixar arquivos, etc. O GitHub expande as funcionalidades básicas oferecidas pelo Git e apresenta uma GUI (do inglês *Graphical User Interface*) intuitiva e amigável, tornando-o bem mais acessível. Utilizou-se também o GitHub Desktop, um aplicativo que possibilita interagir diretamente com o GitHub por meio de uma

interface, sem a necessidade de precisar acessar um navegador *web* ou escrever linhas de comando em um terminal (GITHUB, 2022).

2.3.8 Java

Java é uma linguagem de programação de alto nível e plataforma de computação lançada pela Sun Microsystems em 1995 (posteriormente adquirida pela Oracle Corporation), amplamente utilizada no desenvolvimento de aplicações *web*, *mobile*, *desktop*, empresariais, científicas, etc. (IBM CLOUD EDUCATION, 2019).

Originalmente projetada por James Gosling, Oracle (2019) explica que a linguagem surgiu para atender algumas necessidades, a saber: ser simples e familiar, podendo ser facilmente compreendida e programada pela maioria dos programadores atuais; orientada a objetos, para aproveitar as metodologias modernas de desenvolvimento de *software*; *multithread*, para fornecer alto desempenho em aplicações que requerem a realização de múltiplas tarefas em simultâneo; e interpretada, proporcionando a portabilidade.

Para efetivamente programar em Java, deve-se instalar o *Java Development Kit* (JDK), um ambiente de desenvolvimento usado para criar aplicativos e *applets* em Java. O JDK dispõe de uma série de ferramentas com foco no desenvolvedor, como compilador, depurador e gerador de documentação. Além dessas ferramentas, o JDK contém uma cópia completa do *Java Runtime Environment* (JRE), um pacote que inclui todos os recursos necessários para executar um programa em Java. O JRE, por sua vez, contém o *Java Virtual Machine* (JVM), a plataforma que permite executar o código-fonte (*bytecode*) propriamente em qualquer dispositivo ou sistema operacional (HARTMAN, 2022).

2.3.9 JavaFX

JavaFX ou OpenJFX é uma plataforma de código aberto que viabiliza a construção de aplicações Java com uma interface moderna e acelerada por *hardware*. Os aplicativos desenvolvidos com essa tecnologia são altamente portáteis, podendo ser executados em dispositivos como computadores *desktop*, telefones celulares, TVs e *tablets* (OPENJFX, 2022).

Como relembra Lowe (2014, tradução nossa), para criar uma GUI, inicialmente, usava-se o AWT (*Abstract Window Toolkit*), o kit de ferramentas de *widget* original do Java. Posteriormente, o AWT deu lugar ao Swing, que continha componentes bem mais sofisticados quando comparados ao seu antecessor. Nessa sequência, surge o JavaFX, um *framework* concebido com a proposta de suceder o Swing. Com ele, é possível criar interfaces no estilo de janelas tradicionais que incluem controles familiares, como rótulos, botões, caixas de texto, caixas de seleção, listas suspensas, etc., e adorná-los com efeitos sofisticados, como fontes de luz, perspectiva e animação (por isso o FX em JavaFX, uma grafia fonética para *effects*).

2.3.10 Scene Builder

Como descrito por Vos *et al.* (2017, tradução nossa), a plataforma do JavaFX fornece duas maneiras complementares para criar uma GUI, uma é por meio da criação programática e a outra, declarativa. Em outros termos, significa que o indivíduo pode optar por escrever manualmente todas as linhas de código responsáveis pela sua construção ou fazer uso de um *software* dedicado que irá auxiliá-lo, automatizando e facilitando todo o processo.

O principal *software* do mercado com esse propósito é o JavaFX Scene Builder ou apenas Scene Builder. Desenvolvido pela Gluon e disponível para os sistemas Windows, Mac e Linux, é uma ferramenta de *layout* que possibilita a criação de GUIs em aplicativos JavaFX sem a necessidade de codificação. Nela, os usuários podem arrastar e soltar contêineres, controles e outros componentes visuais em uma área de trabalho, modificar suas propriedades e aplicar folhas de estilo CSS (ORACLE, 2020).

O código do *layout* que está sendo criado é gerado automaticamente em segundo plano e ao final, obtém-se como resultado um arquivo completo em FXML – uma extensão do formato XML (*Extensible Markup Language*) – projetado especificamente para armazenar informações sobre elementos da interface. Ele contém “o quê”, mas não “como” esses elementos devem funcionar, por causa disso é chamado de método declarativo. Por separar o *design* da lógica, os membros de uma equipe podem se concentrar em aspectos específicos do desenvolvimento.

Durante o projeto, o Scene Builder foi utilizado em sua versão 18.0.0, lançada em 31 de março de 2022.

2.3.11 XAMPP

XAMPP é um servidor *web* multiplataforma gratuito e de código aberto desenvolvido pela Apache Friends. Por ser considerado uma pilha de *software* (pacote contendo programas que trabalham em conjunto), inclui o Apache HTTP Server, MariaDB e interpretadores para as linguagens de programação PHP e Perl. Quando executado em uma máquina, o XAMPP atua como um *host* ou servidor local, fornecendo assim um ambiente adequado para testar projetos antes de publicá-los em um servidor remoto (EDUCBA, 2021).

No caso deste trabalho, os seguintes componentes contidos no XAMPP versão 8.1.5-0 foram utilizados com o propósito de realizar a administração do banco de dados:

- a) Apache HTTP Server 2.4.53: um *software* de servidor *web* HTTP multiplataforma de código aberto e gratuito mantido pela ASF (APACHE HTTP SERVER PROJECT, 1997);
- b) MariaDB 10.4.24: segundo MariaDB Foundation (2022), criado pelos desenvolvedores originais do MySQL, trata-se de um sistema de gerenciamento de banco de dados relacional (SGBDR) baseado em SQL (*Structured Query Language*), a linguagem padrão para armazenar, manipular e recuperar dados em bancos de dados;
- c) phpMyAdmin 5.1.3: uma ferramenta escrita em PHP com a finalidade de administrar um servidor MySQL ou MariaDB pela *web* por meio de uma GUI (PHPMYADMIN, 2022). Pode ser usado para criar bancos de dados e tabelas, inserir registros, realizar consultas, manipular contas, entre outras funcionalidades.

3 MODELAGEM

Este capítulo dedica-se à apresentação da modelagem do sistema. Resumidamente, no contexto de um projeto de *software*, é a etapa anterior à codificação, isto é, realizada antes da efetiva implementação em uma linguagem de programação específica. Um modelo é a representação ou simplificação da realidade (neste caso, da academia), que é construído e utilizado para direcionar a implementação subsequente, atuando também como documentação da mesma.

3.1 Diagrama de casos de uso

De acordo com Bezerra (2007, p. 53-54), o “modelo de casos de uso é uma representação das funcionalidades externamente observáveis do sistema e dos elementos externos ao sistema que interagem com ele”. Portanto, “[...] representa os possíveis usos de um sistema, conforme percebidos por um observador externo [...]” (sob o ponto de vista do usuário, por exemplo).

Para criar um modelo de casos de uso, é preciso definir três componentes: casos de uso, atores e relacionamentos. Caso de uso é a “especificação de uma sequência completa de interações entre um sistema e um ou mais agentes externos a esse sistema”, representando assim uma determinada funcionalidade (BEZERRA, 2007, p. 54). Ator é representação de uma entidade externa (pessoa ou dispositivo) que irá interagir com o sistema durante a sua execução; em outros termos, “ator é qualquer coisa que se comunica com o sistema ou o produto e que é externa ao sistema em si” (PRESSMAN, 2011, p. 138). Por último, o relacionamento diz respeito à ligação existente entre os casos de uso e os atores, podendo acontecer, como descrito por Bezerra (2007), de quatro maneiras:

- a) comunicação: informa a que caso de uso o ator está associado; é o tipo de relacionamento mais utilizado;
- b) inclusão (*include*): quando um caso de uso (base) inclui a funcionalidade de outro caso de uso (inclusão), significa que sempre que o primeiro for executado, o segundo também será;
- c) extensão (*extends*): quando um caso de uso (extensão) estende a funcionalidade de outro caso de uso (base), significa que quando o

segundo for executado, o comportamento do primeiro pode ou não ocorrer (é eventual);

- d) generalização: permite que um caso de uso (ou ator) herde características de outro mais genérico.

A partir desse modelo, com o auxílio de todos os componentes citados, um desenvolvedor consegue estabelecer o comportamento prévio do sistema e descrever todos os requisitos funcionais especificados durante a etapa de levantamento de requisitos (item 2.2.1). Para representá-lo, utiliza-se uma das ferramentas definidas pela UML, o diagrama de casos de uso:

Os casos de uso são documentados por um diagrama de casos de uso de alto nível. O conjunto de casos de uso representa todas as possíveis interações que serão descritas nos requisitos de sistema. Atores, que podem ser pessoas ou outros sistemas, são representados como figuras 'palito'. Cada classe de interação é representada por uma elipse. Linhas fazem a ligação entre os atores e a interação. Opcionalmente, pontas de flechas podem ser adicionadas às linhas para mostrar como a interação se inicia (SOMMERVILLE, 2011, p. 74).

Esta situação – o diagrama de casos de uso para o sistema proposto neste trabalho – está ilustrada na Figura 1.

Figura 1 – Diagrama de casos de uso



Fonte: Elaboração própria (2022).

Além disso, cada caso de uso identificado no diagrama foi definido utilizando uma descrição narrativa (textual), detalhando as interações que ocorrem. Bezerra (2007, p. 55) comenta que a “UML não define uma estrutura textual a ser utilizada na descrição de um caso de uso. Consequentemente, há vários estilos de descrição propostos para definir casos de uso”. Como sugestão, foram descritos usando um formato tabular. Desse modo, o quadro é dividido em duas colunas: uma coluna contém as ações do usuário; a outra, as ações do sistema. Para compreender esse modelo, a leitura deve ser feita em ziguezague, respeitando a ordem numérica. O grau de detalhamento utilizado em cada caso de uso foi relativamente sucinto (especialmente para os casos de uso mais complexos), ou seja, não descreve as interações entre ator e sistema com todos os detalhes. Os casos de uso dos módulos desenvolvidos serão exibidos nos Quadros 4 a 22.

Quadro 4 – Descrição do caso de uso "Realizar cadastro de usuário"

Nome do caso de uso	Realizar cadastro de usuário.
Ator	Usuário não autorizado.
Objetivo	Este caso de uso descreve as etapas percorridas por um usuário para se cadastrar no sistema.
Prioridade	Desejável.
Ações do ator	Ações do sistema
1. Inserir o nome de usuário no campo "Usuário", a senha no campo "Senha" e "Confirmar Senha", e clicar no botão "Cadastrar".	
	2. Obter nome de usuário, senha e confirmação de senha (conforme V01, V02 e V03) e aplicar o algoritmo criptográfico SHA-256 nas senhas.
	3. Inserir o usuário no banco de dados, exibir mensagem de sucesso e redirecionar o usuário para a tela de <i>login</i> .
Validações	V01: os campos não podem estar vazios; V02: a senha e a confirmação de senha devem ser iguais; V03: não pode haver um cadastro com aquele nome de usuário.

Fonte: Elaboração própria (2022).

Quadro 5 – Descrição do caso de uso "Efetuar login"

Nome do caso de uso	Efetuar <i>login</i> .
Ator	Usuário não autorizado.
Objetivo	Este caso de uso descreve as etapas percorridas por um usuário para entrar no sistema.
Prioridade	Desejável.
Ações do ator	Ações do sistema
1. Inserir o nome de usuário no campo "Usuário", a senha no campo "Senha" e clicar no botão "Entrar".	
	2. Obter nome de usuário e senha (conforme V01) e aplicar o algoritmo criptográfico SHA-256 na senha.
	3. Comparar as credenciais (nome de usuário e senha criptografada) com as cadastradas no banco de dados (conforme V02 e V03).
	4. Autenticar e redirecionar o usuário para a tela inicial do sistema.
Validações	V01: os campos não podem estar vazios; V02: o usuário deve estar cadastrado no banco; V03: as credenciais digitadas devem coincidir com as cadastradas.

Fonte: Elaboração própria (2022).

Quadro 6 – Descrição do caso de uso "Manter fornecedores"

Nome do caso de uso	Manter fornecedores.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, atualizar, excluir ou consultar fornecedores.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Administrativo" e depois no item de menu "Fornecedores".	
	2. Redirecionar o usuário para a tela de gerenciamento de fornecedores e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, atualizar, excluir ou consultar um fornecedor.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01, V02 e V03) e inserir no banco de dados.
7. Para atualizar um fornecedor, o usuário deve selecioná-lo na tabela e clicar em "Atualizar".	
	8. Abrir a janela de atualização e carregar os dados previamente cadastrados.
9. Informar os dados a serem atualizados e clicar no botão "Salvar".	
	10. Obter os dados (conforme V01 e V02) e atualizar no banco de dados.
11. Para excluir um fornecedor, o usuário deve selecioná-lo na tabela e clicar em "Excluir".	
	12. Excluir o fornecedor do banco de dados.
13. Para consultar um fornecedor, o usuário deve informar os parâmetros na barra de pesquisa.	
	14. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos; V03: o fornecedor não pode estar cadastrado.

Fonte: Elaboração própria (2022).

Quadro 7 – Descrição do caso de uso "Manter produtos"

Nome do caso de uso	Manter produtos.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, atualizar, excluir ou consultar produtos.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Vendas" e depois no item de menu "Produtos".	
	2. Redirecionar o usuário para a tela de gerenciamento de produtos e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, atualizar, excluir ou consultar um produto.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01 e V02) e inserir no banco de dados.
7. Para atualizar um produto, o usuário deve selecioná-lo na tabela e clicar em "Atualizar".	
	8. Abrir a janela de atualização e carregar os dados previamente cadastrados.
9. Informar os dados a serem atualizados e clicar no botão "Salvar".	
	10. Obter os dados (conforme V01 e V02) e atualizar no banco de dados.
11. Para excluir um produto, o usuário deve selecioná-lo na tabela e clicar em "Excluir".	
	12. Excluir o produto do banco de dados.
13. Para consultar um produto, o usuário deve informar os parâmetros na barra de pesquisa.	
	14. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos.

Fonte: Elaboração própria (2022).

Quadro 8 – Descrição do caso de uso "Manter entradas de estoque"

Nome do caso de uso	Manter entradas de estoque.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, cancelar ou consultar entradas de estoque.
Prioridade	Desejável.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Vendas" e depois no item de menu "Entradas de Estoque".	
	2. Redirecionar o usuário para a tela de gerenciamento de entradas de estoque e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, cancelar ou consultar uma entrada de estoque.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados, adicionar uma lista de produtos à entrada e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01 e V02) e a lista de produtos e inserir no banco de dados.
7. Para cancelar uma entrada, o usuário deve selecioná-la na tabela e clicar em "Cancelar".	
	8. Cancelar a entrada de estoque e a lista de produtos do banco de dados.
9. Para consultar uma entrada, o usuário deve informar os parâmetros na barra de pesquisa.	
	10. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos.

Fonte: Elaboração própria (2022).

Quadro 9 – Descrição do caso de uso "Manter clientes"

Nome do caso de uso	Manter clientes.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, atualizar, excluir ou consultar clientes.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Administrativo" e depois no item de menu "Clientes".	
	2. Redirecionar o usuário para a tela de gerenciamento de clientes e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, atualizar, excluir ou consultar um cliente.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01, V02 e V03) e inserir no banco de dados.
7. Para atualizar um cliente, o usuário deve selecioná-lo na tabela e clicar em "Atualizar".	
	8. Abrir a janela de atualização e carregar os dados previamente cadastrados.
9. Informar os dados a serem atualizados e clicar no botão "Salvar".	
	10. Obter os dados (conforme V01 e V02) e atualizar no banco de dados.
11. Para excluir um cliente, o usuário deve selecioná-lo na tabela e clicar em "Excluir".	
	12. Excluir o cliente do banco de dados.
13. Para consultar um cliente, o usuário deve informar os parâmetros na barra de pesquisa.	
	14. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos; V03: o cliente não pode estar cadastrado.

Fonte: Elaboração própria (2022).

Quadro 10 – Descrição do caso de uso "Manter vendas"

Nome do caso de uso	Manter vendas.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, cancelar ou consultar vendas.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Vendas" e depois no item de menu "Vendas".	
	2. Redirecionar o usuário para a tela de gerenciamento de vendas e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, cancelar ou consultar uma venda.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados, adicionar uma lista de produtos à venda e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01 e V02) e a lista de produtos e inserir no banco de dados.
7. Para cancelar uma venda, o usuário deve selecioná-la na tabela e clicar em "Cancelar".	
	8. Cancelar a venda e a lista de produtos do banco de dados.
9. Para consultar uma venda, o usuário deve informar os parâmetros na barra de pesquisa.	
	10. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos.

Fonte: Elaboração própria (2022).

Quadro 11 – Descrição do caso de uso "Manter exercícios"

Nome do caso de uso	Manter exercícios.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, atualizar, excluir ou consultar exercícios.
Prioridade	Desejável.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Treinos" e depois no item de menu "Exercícios".	
	2. Redirecionar o usuário para a tela de gerenciamento de exercícios e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, atualizar, excluir ou consultar um exercício.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01 e V02) e inserir no banco de dados.
7. Para atualizar um exercício, o usuário deve selecioná-lo na tabela e clicar em "Atualizar".	
	8. Abrir a janela de atualização e carregar os dados previamente cadastrados.
9. Informar os dados a serem atualizados e clicar no botão "Salvar".	
	10. Obter os dados (conforme V01 e V02) e atualizar no banco de dados.
11. Para excluir um exercício, o usuário deve selecioná-lo na tabela e clicar em "Excluir".	
	12. Excluir o exercício do banco de dados.
13. Para consultar um exercício, o usuário deve informar os parâmetros na barra de pesquisa.	
	14. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos.

Fonte: Elaboração própria (2022).

Quadro 12 – Descrição do caso de uso "Manter treinos"

Nome do caso de uso	Manter treinos.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, excluir ou consultar treinos.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Treinos" e depois no item de menu "Treinos".	
	2. Redirecionar o usuário para a tela de gerenciamento de treinos e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, excluir ou consultar um treino.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados, adicionar uma lista de exercícios ao treino e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01 e V02) e a lista de exercícios e inserir no banco de dados.
7. Para excluir um treino, o usuário deve selecioná-lo na tabela e clicar em "Excluir".	
	8. Excluir o treino e a lista de exercícios do banco de dados.
9. Para consultar um treino, o usuário deve informar os parâmetros na barra de pesquisa.	
	10. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos.

Fonte: Elaboração própria (2022).

Quadro 13 – Descrição do caso de uso "Manter fichas"

Nome do caso de uso	Manter fichas.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, excluir ou consultar fichas.
Prioridade	Desejável.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Treinos" e depois no item de menu "Fichas".	
	2. Redirecionar o usuário para a tela de gerenciamento de fichas e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, excluir ou consultar uma ficha.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados, adicionar uma lista de treinos à ficha e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01 e V02) e a lista de treinos e inserir no banco de dados.
7. Para excluir uma ficha, o usuário deve selecioná-la na tabela e clicar em "Excluir".	
	8. Excluir a ficha e a lista de treinos do banco de dados.
9. Para consultar uma ficha, o usuário deve informar os parâmetros na barra de pesquisa.	
	10. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos.

Fonte: Elaboração própria (2022).

Quadro 14 – Descrição do caso de uso "Manter medidas"

Nome do caso de uso	Manter medidas.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, atualizar, excluir ou consultar medidas dos perímetros corporais de clientes.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Avaliações" e depois no item de menu "Medidas".	
	2. Redirecionar o usuário para a tela de gerenciamento de medidas e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, atualizar, excluir ou consultar as medidas de um cliente.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01 e V02) e inserir no banco de dados.
7. Para atualizar a medição de um cliente, o usuário deve selecioná-la na tabela e clicar em "Atualizar".	
	8. Abrir a janela de atualização e carregar os dados previamente cadastrados.
9. Informar os dados a serem atualizados e clicar no botão "Salvar".	
	10. Obter os dados (conforme V01 e V02) e atualizar no banco de dados.
11. Para excluir a medição de um cliente, o usuário deve selecioná-la na tabela e clicar em "Excluir".	
	12. Excluir as medidas do cliente do banco de dados.
13. Para consultar as medidas de um cliente, o usuário deve informar os parâmetros na barra de pesquisa.	
	14. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos.

Fonte: Elaboração própria (2022).

Quadro 15 – Descrição do caso de uso "Manter mensalidades"

Nome do caso de uso	Manter mensalidades.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar, cadastrar, atualizar, cancelar ou consultar mensalidades.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Administrativo" e depois no item de menu "Mensalidades".	
	2. Redirecionar o usuário para a tela de gerenciamento de mensalidades e carregar os dados cadastrados na tabela de visualização.
3. Decidir se deseja cadastrar, atualizar, cancelar ou consultar uma mensalidade.	
	4. Se a escolha for cadastrar, abrir a janela de cadastro.
5. Informar os dados a serem cadastrados e clicar no botão "Salvar".	
	6. Obter os dados (conforme V01 e V02) e inserir no banco de dados.
7. Para atualizar uma mensalidade, o usuário deve selecioná-la na tabela e clicar em "Atualizar".	
	8. Abrir a janela de atualização e carregar os dados previamente cadastrados.
9. Informar os dados a serem atualizados e clicar no botão "Salvar".	
	10. Obter os dados (conforme V01 e V02) e atualizar no banco de dados.
11. Para cancelar uma mensalidade, o usuário deve selecioná-la na tabela e clicar em "Cancelar".	
	12. Cancelar a mensalidade do banco de dados.
13. Para consultar uma mensalidade, o usuário deve informar os parâmetros na barra de pesquisa.	
	14. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	V01: os campos obrigatórios não podem estar vazios; V02: os dados não podem ser inválidos.

Fonte: Elaboração própria (2022).

Quadro 16 – Descrição do caso de uso "Manter cobranças"

Nome do caso de uso	Manter cobranças.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar e consultar as cobranças geradas.
Prioridade	Importante.
Ações do ator	Ações do sistema
1. Cadastrar uma venda ou mensalidade.	
	2. Gerar uma cobrança referente à ação tomada e cadastrá-la no banco de dados.
3. Para visualizar as cobranças geradas, clicar na opção de menu "Financeiro" e depois no item de menu "Cobranças".	
	4. Redirecionar o usuário para a tela de gerenciamento de cobranças e carregar os dados cadastrados na tabela de visualização.
5. Para consultar uma cobrança, o usuário deve informar os parâmetros na barra de pesquisa.	
	6. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	Não possui.

Fonte: Elaboração própria (2022).

Quadro 17 – Descrição do caso de uso "Receber cobranças"

Nome do caso de uso	Receber cobranças.
Ator	Usuário autorizado.
Objetivo	Permitir que o usuário possa receber o pagamento das cobranças.
Prioridade	Importante.
Ações do ator	Ações do sistema
1. Selecionar uma cobrança na tabela e clicar no botão "Receber Cobrança".	
	2. Abrir a janela de recebimento.
3. Informar o valor a ser pago e clicar no botão "Salvar".	
	4. Obter valor a ser pago, dar baixa na cobrança, cadastrar pagamento e exibir mensagem de sucesso.
Validações	Não possui.

Fonte: Elaboração própria (2022).

Quadro 18 – Descrição do caso de uso "Manter pagamentos"

Nome do caso de uso	Manter pagamentos.
Ator	Usuário autorizado.
Objetivo	Permitir que um usuário possa visualizar e consultar os pagamentos recebidos.
Prioridade	Importante.
Ações do ator	Ações do sistema
1. Na tela principal do sistema, clicar na opção de menu "Financeiro" e depois no item de menu "Pagamentos".	
	2. Redirecionar o usuário para a tela de gerenciamento de pagamentos e carregar os dados cadastrados na tabela de visualização.
3. Para consultar um pagamento, o usuário deve informar os parâmetros na barra de pesquisa.	
	4. Filtrar a tabela de visualização exibindo somente os registros que atendem esses parâmetros.
Validações	Não possui.

Fonte: Elaboração própria (2022).

Quadro 19 – Descrição do caso de uso "Visualizar produtos inseridos"

Nome do caso de uso	Visualizar produtos inseridos.
Ator	Usuário autorizado.
Objetivo	Permitir que o usuário possa visualizar os produtos inseridos em uma entrada de estoque ou venda.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Selecionar uma entrada de estoque ou venda na tabela e clicar no botão "Listar Produtos".	
	2. Abrir uma janela contendo uma tabela que possui todos os produtos que foram inseridos naquela entrada de estoque ou venda selecionada.
Validações	Não possui.

Fonte: Elaboração própria (2022).

Quadro 20 – Descrição do caso de uso "Visualizar exercícios inseridos"

Nome do caso de uso	Visualizar exercícios inseridos.
Ator	Usuário autorizado.
Objetivo	Permitir que o usuário possa visualizar os exercícios inseridos em um treino.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Selecionar um treino na tabela e clicar no botão "Listar Exercícios".	
	2. Abrir uma janela contendo uma tabela que possui todos os exercícios que foram inseridos naquele treino selecionado.
Validações	Não possui.

Fonte: Elaboração própria (2022).

Quadro 21 – Descrição do caso de uso "Visualizar treinos inseridos"

Nome do caso de uso	Visualizar treinos inseridos.
Ator	Usuário autorizado.
Objetivo	Permitir que o usuário possa visualizar os treinos inseridos em uma ficha.
Prioridade	Essencial.
Ações do ator	Ações do sistema
1. Selecionar uma ficha na tabela e clicar no botão "Listar Treinos".	
	2. Abrir uma janela contendo uma tabela que possui todos os treinos que foram inseridos naquela ficha selecionada.
Validações	Não possui.

Fonte: Elaboração própria (2022).

Quadro 22 – Descrição do caso de uso "Visualizar *dashboard*"

Nome do caso de uso	Visualizar <i>dashboard</i> .
Ator	Usuário autorizado.
Objetivo	Exibir uma <i>dashboard</i> na tela inicial do sistema.
Prioridade	Importante.
Ações do ator	Ações do sistema
	1. Carregar informações na <i>dashboard</i> e exibi-la na tela inicial do sistema.
Validações	Não possui.

Fonte: Elaboração própria (2022).

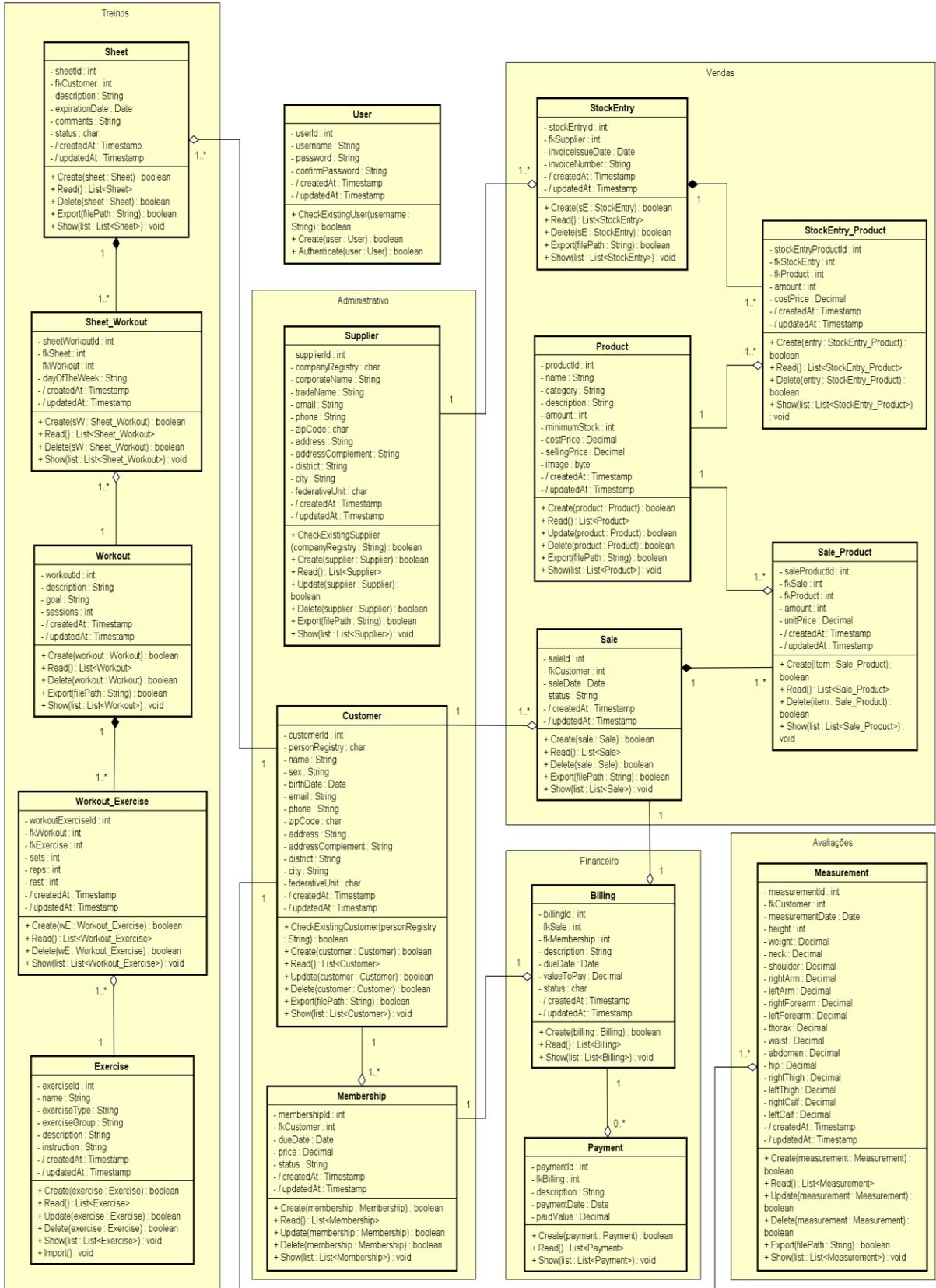
3.2 Diagrama de classes

Considerado um dos diagramas mais importantes da UML, o diagrama de classes, nas palavras de Sommerville (2011, p. 90), é utilizado no “desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes de um sistema e as associações entre essas classes”. Isso significa que ao fazer uso desse recurso de modelagem, obtém-se uma visão estrutural do sistema, auxiliando em sua futura implementação prática.

Quanto aos elementos principais desse diagrama, em uma analogia feita por Bezerra (2007), uma classe é representada por uma “caixa” dividida em, no máximo, três compartimentos/partes horizontais. O compartimento superior contém o nome da classe. O compartimento do meio lista os atributos, que correspondem às informações armazenadas pelos objetos da classe. Por fim, o compartimento inferior declara as operações (também chamadas de métodos), que representam as ações que podem ser executadas por um objeto. Além disso, Guedes (2011) destaca a existência dos relacionamentos entre as classes, conhecidos por associações, que permitem que elas compartilhem informações entre si. Uma associação é simbolizada por uma linha que liga as classes envolvidas.

Para documentar a estrutura do sistema deste projeto, a equipe criou o diagrama de classes que pode ser visualizado na Figura 2.

Figura 2 – Diagrama de classes



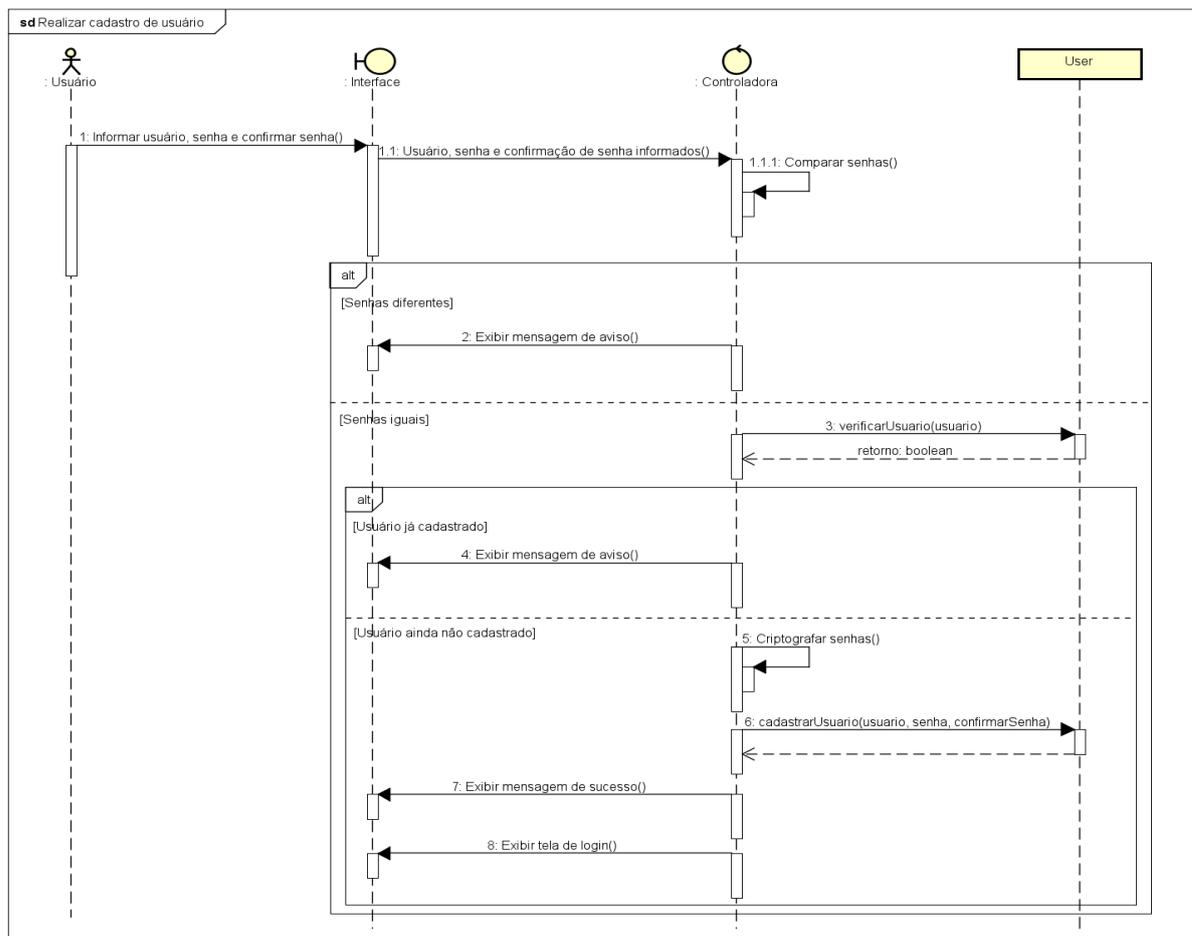
Fonte: Elaboração própria (2022).

3.3 Diagrama de sequência

O diagrama de sequência é mais um recurso da UML que foi utilizado neste projeto com o intuito de documentá-lo. Nas palavras de Guedes (2011, p. 192), esse diagrama “[...] procura determinar a sequência de eventos que ocorrem em um determinado processo, identificando quais mensagens devem ser disparadas entre os elementos envolvidos e em que ordem”. Segundo o autor, o diagrama de sequência é fundamentado no diagrama de casos de uso, de forma a existir um diagrama de sequência para cada caso de uso descrito no sistema. Além disso, também possui dependência do diagrama de classes, já que faz uso das classes declaradas nele. Com base nisso, as Figuras 3 a 21 exibem os diagramas de sequência elaborados.

A Figura 3 descreve as etapas percorridas por um usuário para se cadastrar no sistema.

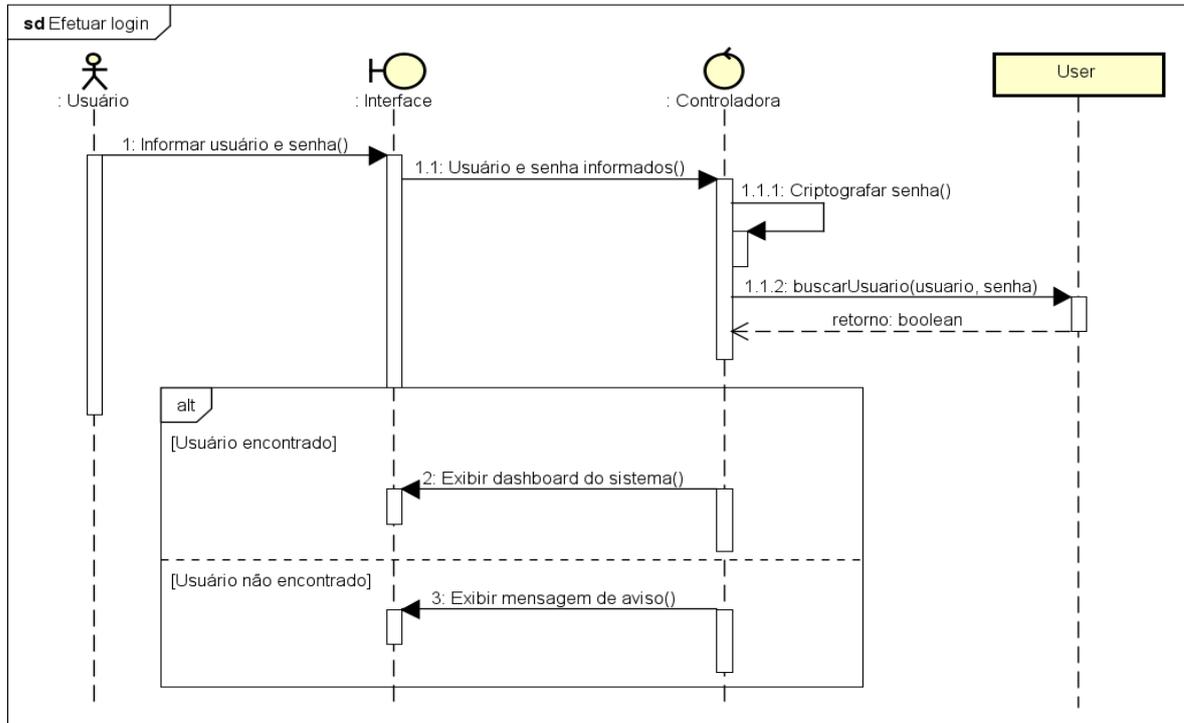
Figura 3 – Diagrama de sequência "Realizar cadastro de usuário"



Fonte: Elaboração própria (2022).

A Figura 4 descreve as etapas percorridas por um usuário para entrar no sistema.

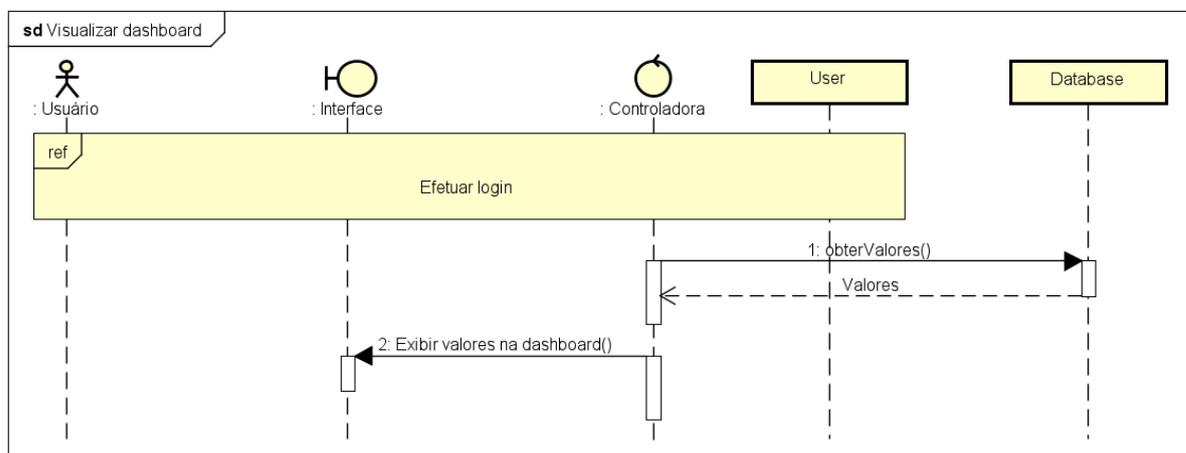
Figura 4 – Diagrama de sequência "Efetuar login"



Fonte: Elaboração própria (2022).

A Figura 5 é responsável por apresentar os passos necessários ao acesso e carregamento da *dashboard* do sistema.

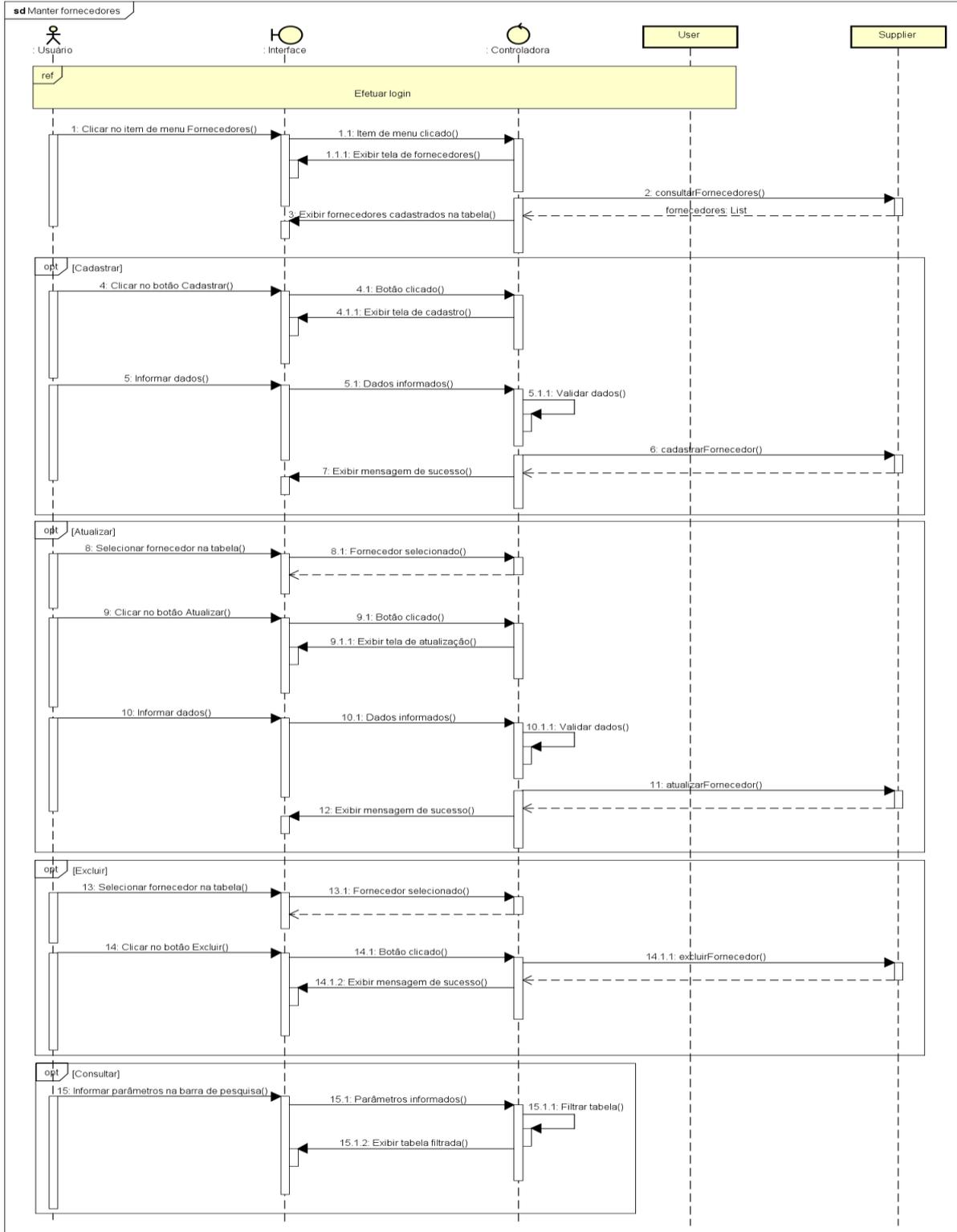
Figura 5 – Diagrama de sequência "Visualizar dashboard"



Fonte: Elaboração própria (2022).

A Figura 6 apresenta a seqüência de eventos que permitem com que o usuário possa visualizar, cadastrar, atualizar, excluir ou consultar fornecedores.

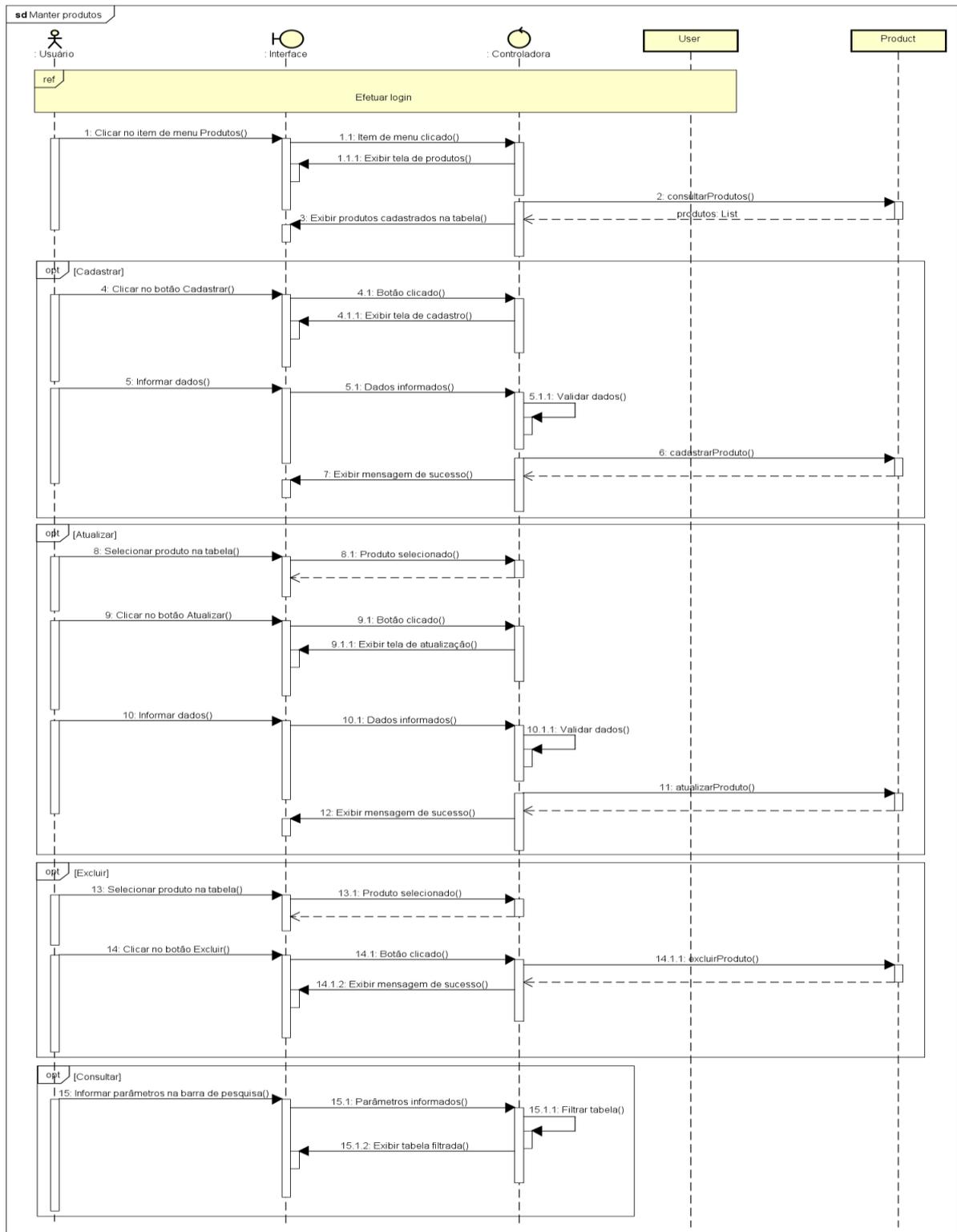
Figura 6 – Diagrama de seqüência "Manter fornecedores"



Fonte: Elaboração própria (2022).

A Figura 7 apresenta a sequência de eventos que permitem com que o usuário possa visualizar, cadastrar, atualizar, excluir ou consultar produtos.

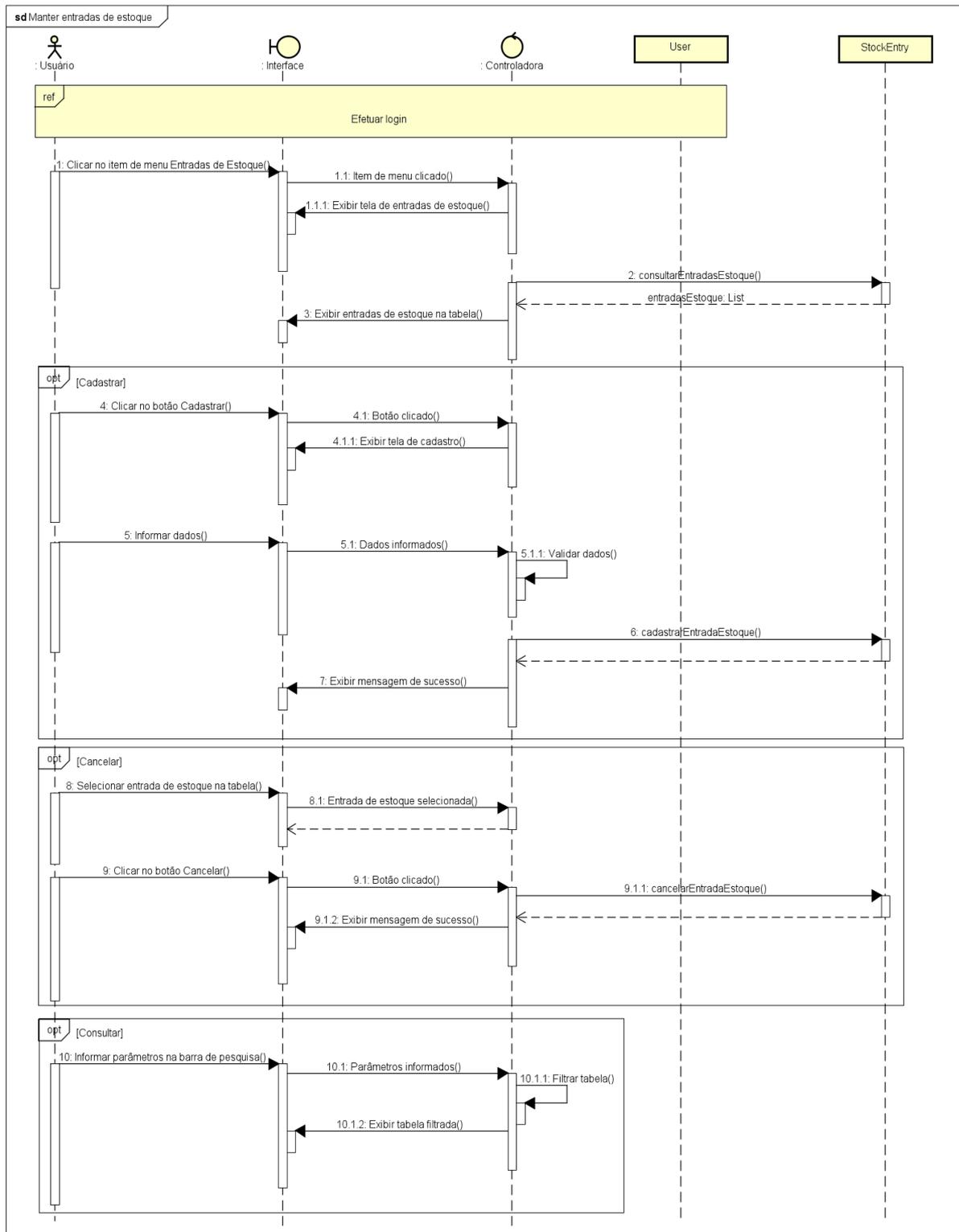
Figura 7 – Diagrama de sequência "Manter produtos"



Fonte: Elaboração própria (2022).

A Figura 8 apresenta a seqüência de eventos que permitem com que o usuário possa visualizar, cadastrar, cancelar ou consultar entradas de estoque.

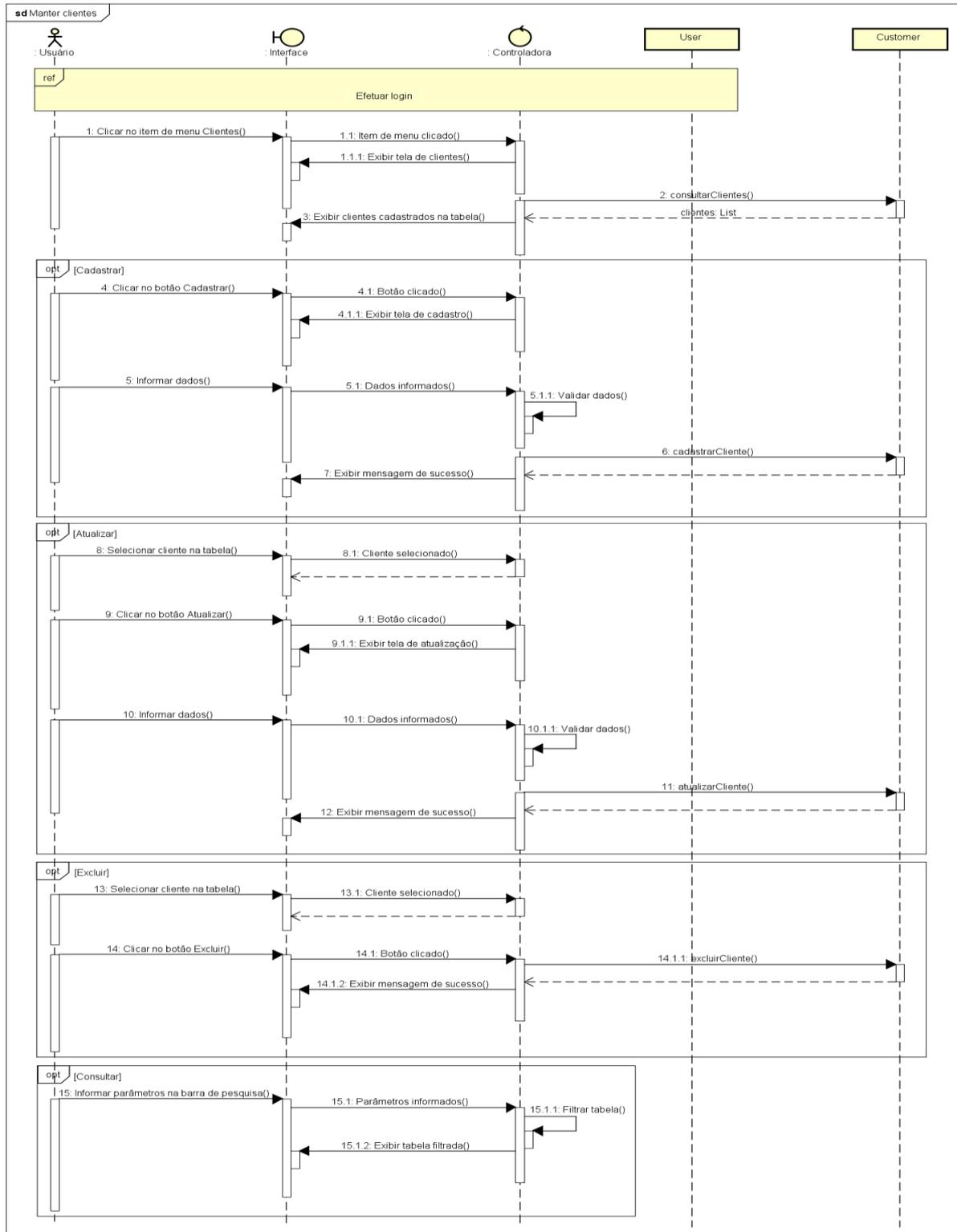
Figura 8 – Diagrama de seqüência "Manter entradas de estoque"



Fonte: Elaboração própria (2022).

A Figura 9 apresenta a sequência de eventos que permitem com que o usuário possa visualizar, cadastrar, atualizar, excluir ou consultar clientes.

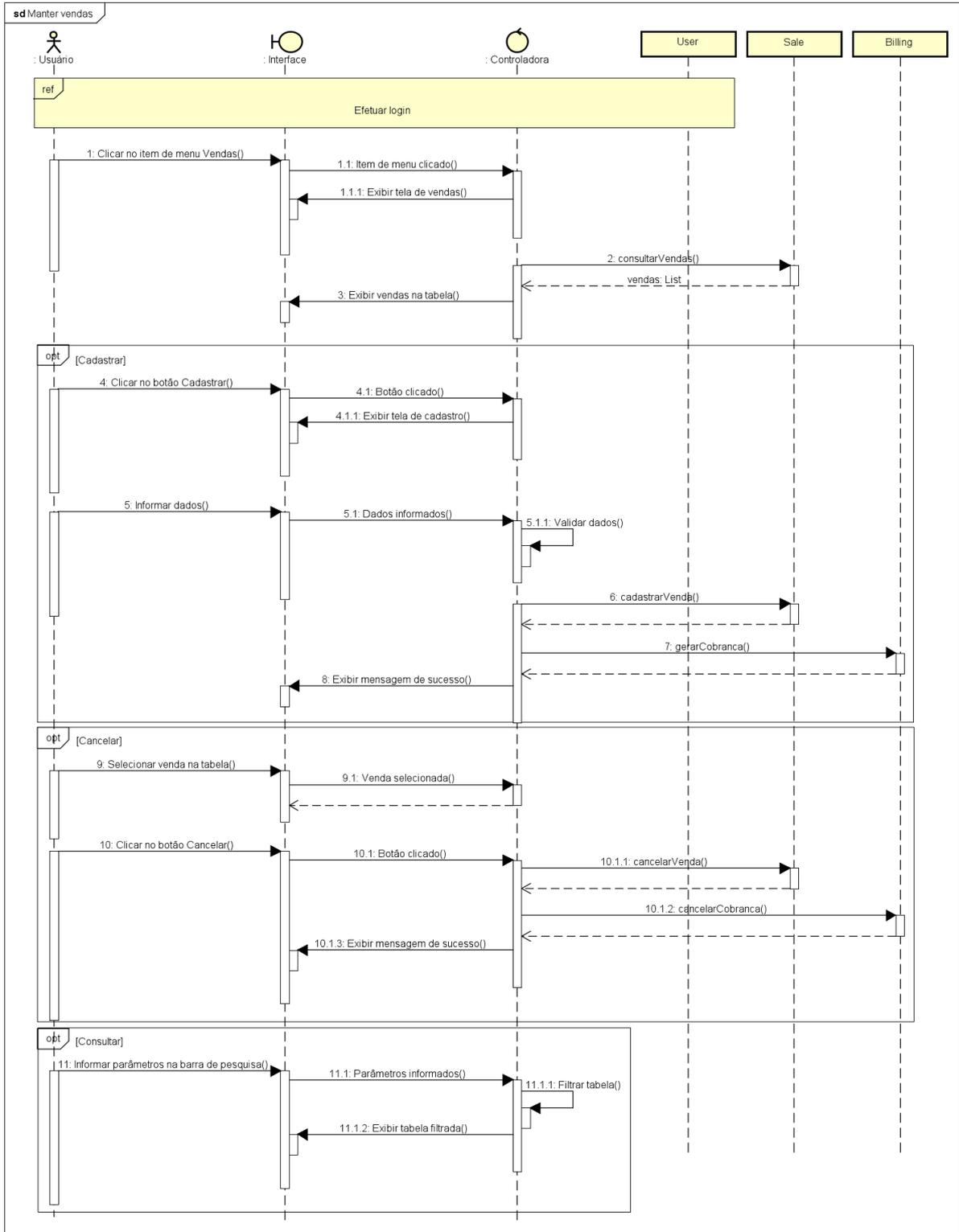
Figura 9 – Diagrama de sequência "Manter clientes"



Fonte: Elaboração própria (2022).

A Figura 10 apresenta a sequência de eventos que permitem com que o usuário possa visualizar, cadastrar, cancelar ou consultar vendas.

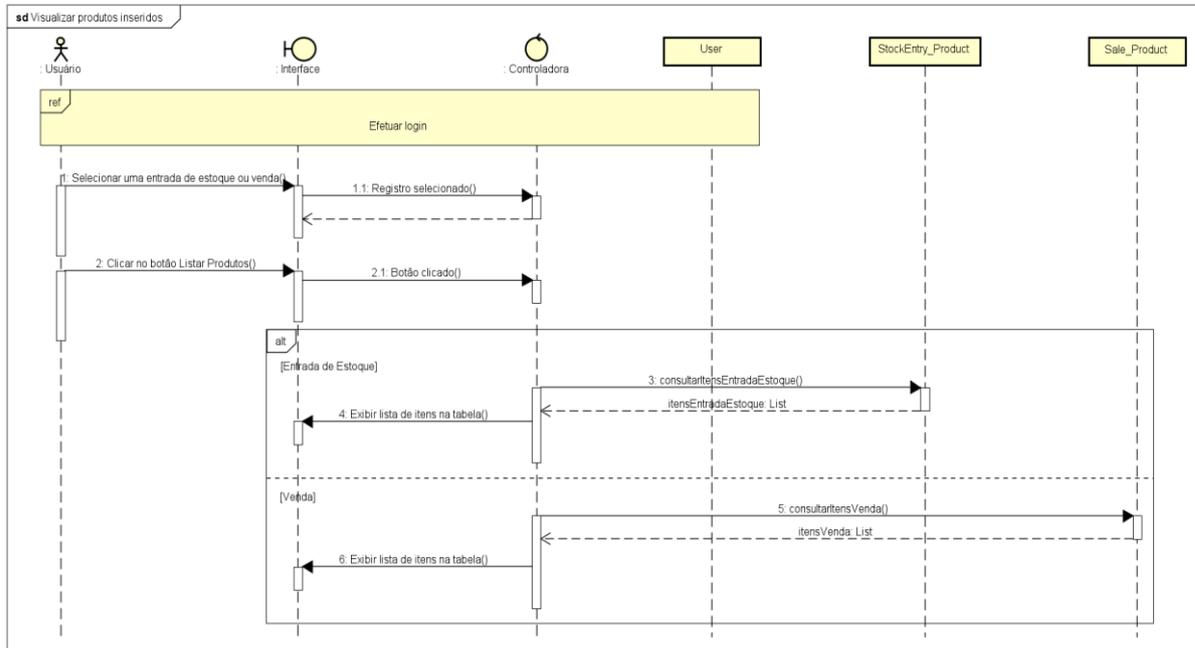
Figura 10 – Diagrama de sequência "Manter vendas"



Fonte: Elaboração própria (2022).

A Figura 11 apresenta a sequência de eventos que permitem com que o usuário possa visualizar os produtos inseridos em uma determinada entrada de estoque ou venda.

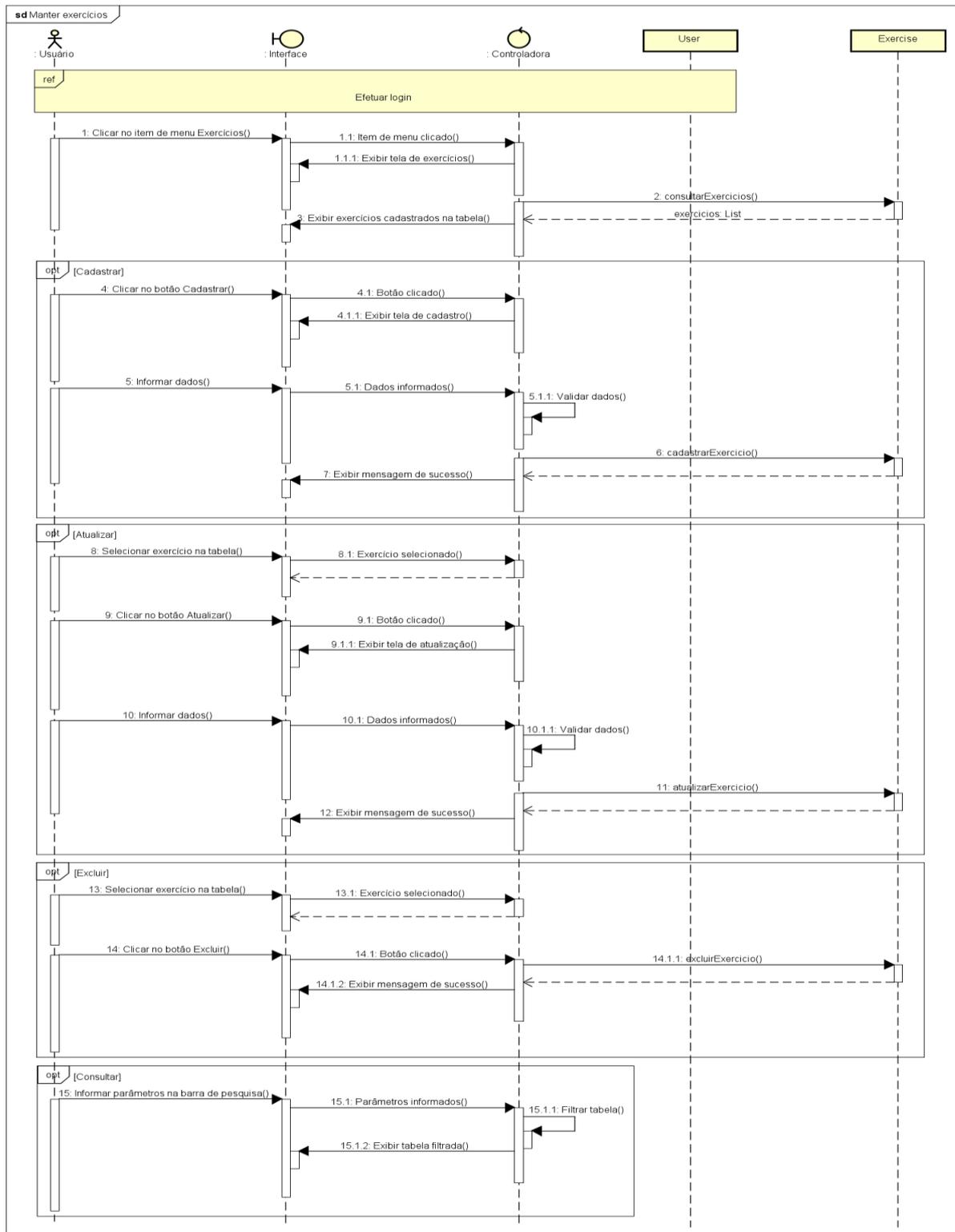
Figura 11 – Diagrama de sequência "Visualizar produtos inseridos"



Fonte: Elaboração própria (2022).

A Figura 12 apresenta a sequência de eventos que permitem com que o usuário possa visualizar, cadastrar, atualizar, excluir ou consultar exercícios.

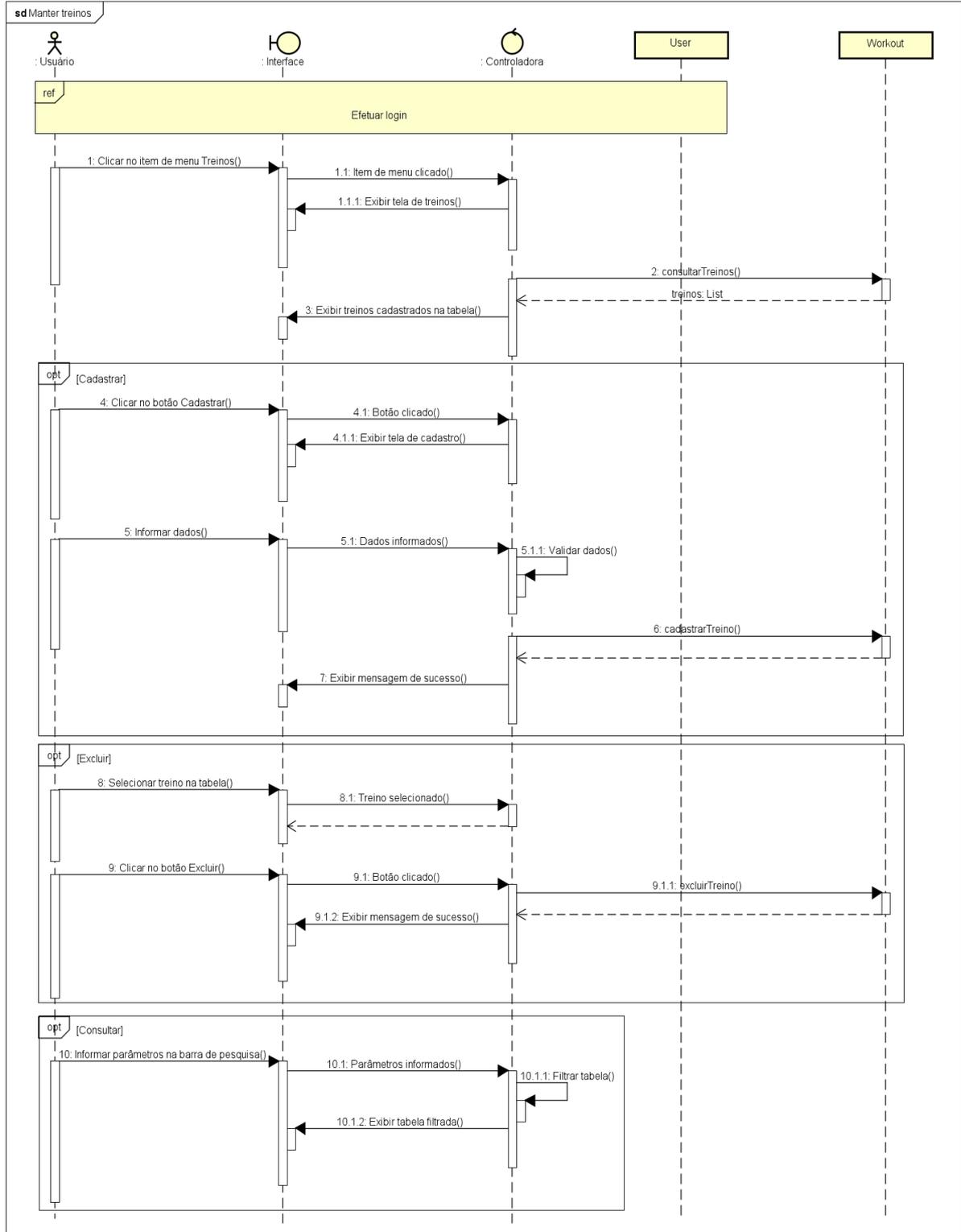
Figura 12 – Diagrama de sequência "Manter exercícios"



Fonte: Elaboração própria (2022).

A Figura 13 apresenta a sequência de eventos que permitem com que o usuário possa visualizar, cadastrar, excluir ou consultar treinos.

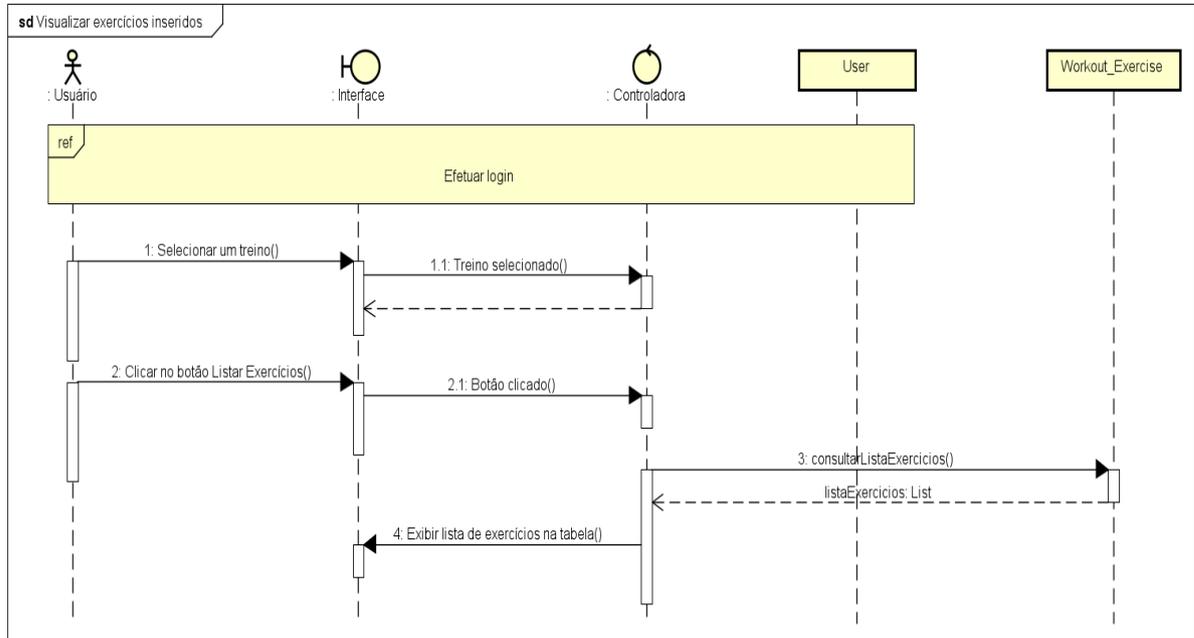
Figura 13 – Diagrama de sequência "Manter treinos"



Fonte: Elaboração própria (2022).

A Figura 14 apresenta a sequência de eventos que permitem com que o usuário possa visualizar os exercícios inseridos em um determinado treino.

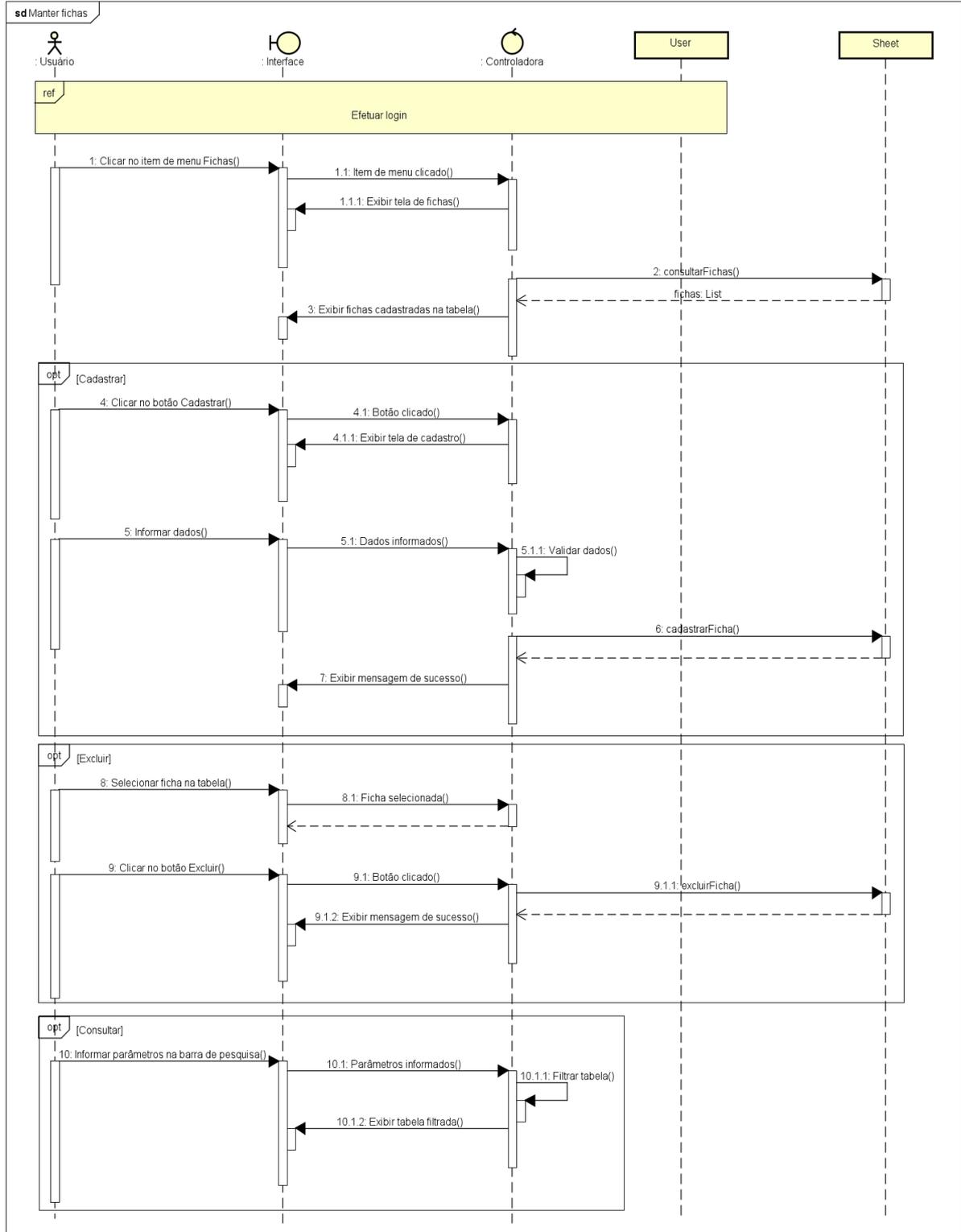
Figura 14 – Diagrama de sequência "Visualizar exercícios inseridos"



Fonte: Elaboração própria (2022).

A Figura 15 apresenta a sequência de eventos que permitem com que o usuário possa visualizar, cadastrar, excluir ou consultar fichas.

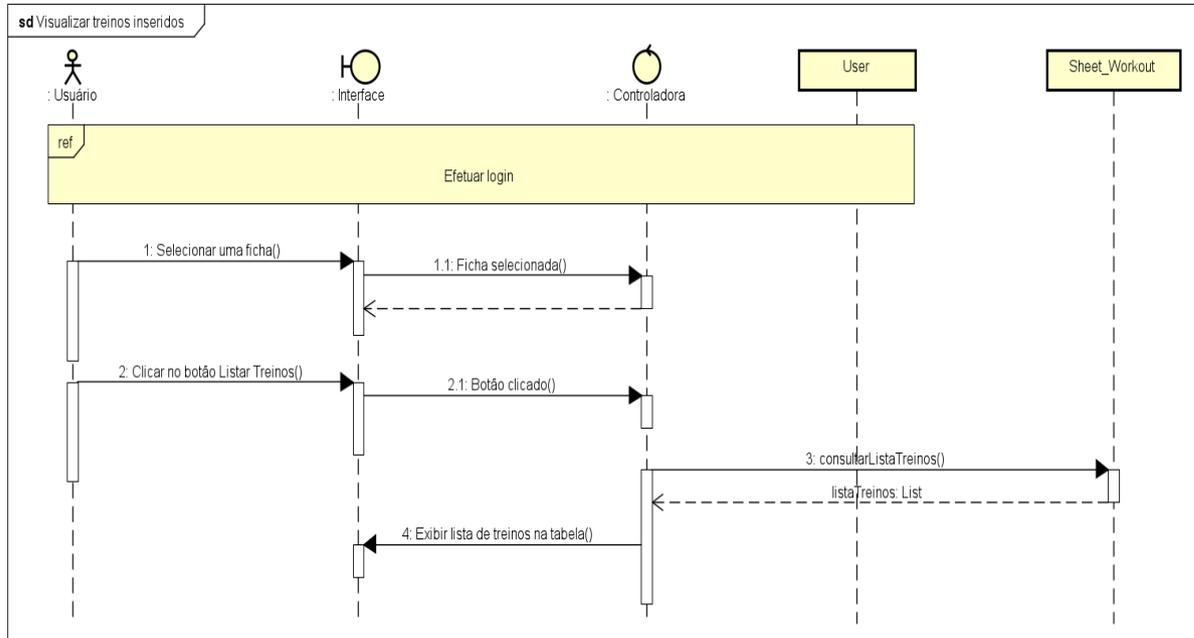
Figura 15 – Diagrama de sequência "Manter fichas"



Fonte: Elaboração própria (2022).

A Figura 16 apresenta a sequência de eventos que permitem com que o usuário possa visualizar os treinos inseridos em uma determinada ficha.

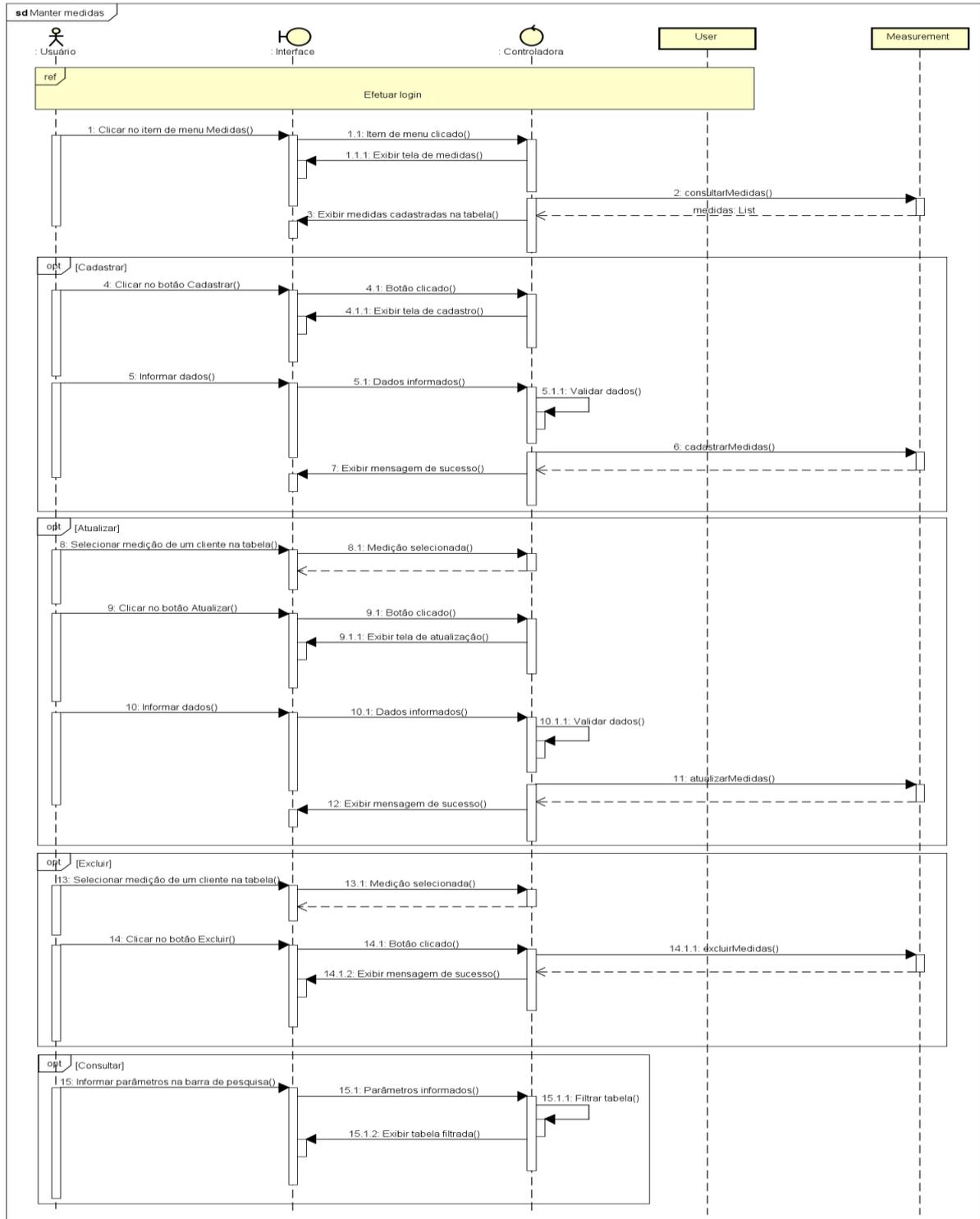
Figura 16 – Diagrama de sequência "Visualizar treinos inseridos"



Fonte: Elaboração própria (2022).

A Figura 17 apresenta a sequência de eventos que permitem com que o usuário possa visualizar, cadastrar, atualizar, excluir ou consultar as medidas dos perímetros corporais de clientes.

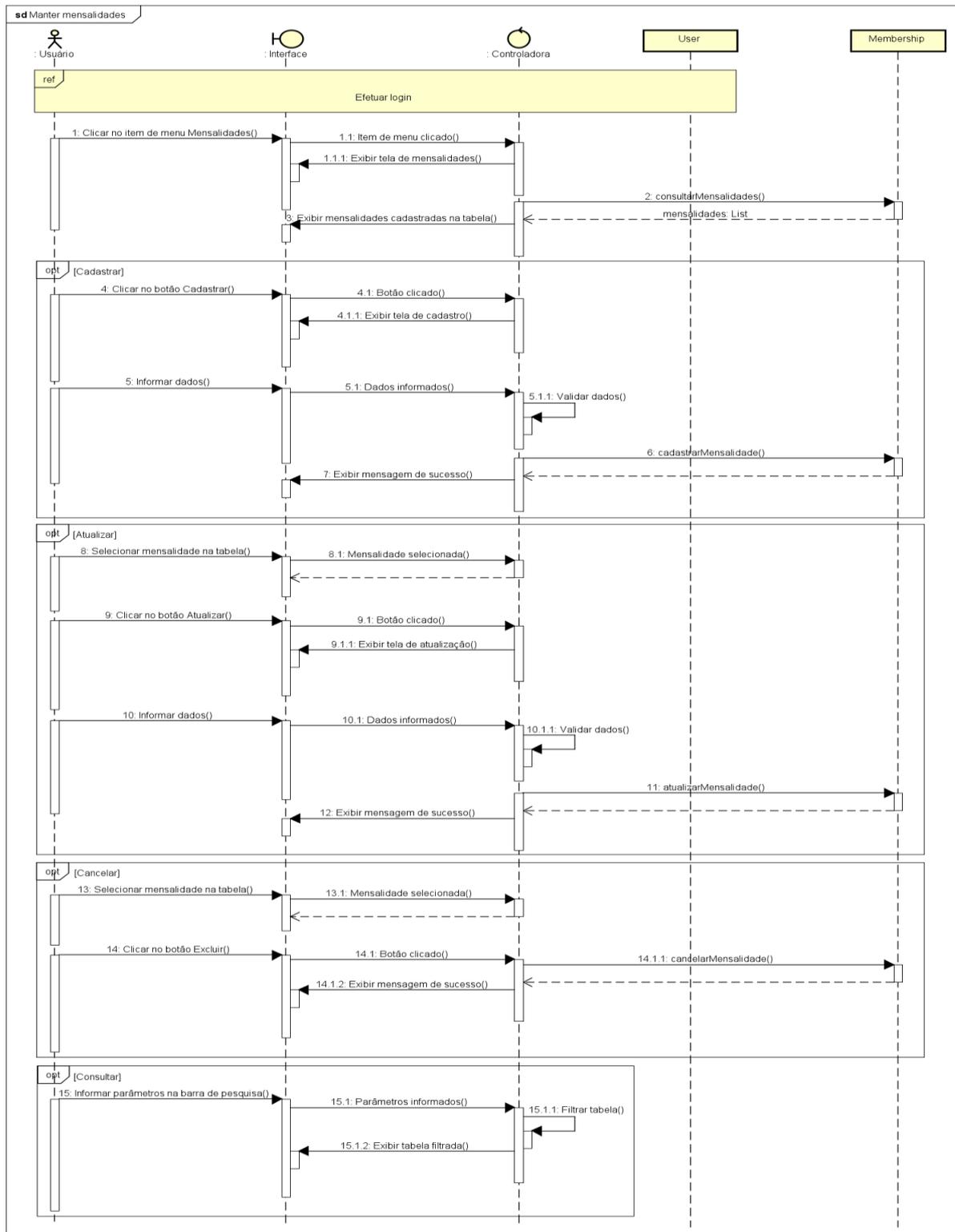
Figura 17 – Diagrama de sequência "Manter medidas"



Fonte: Elaboração própria (2022).

A Figura 18 apresenta a sequência de eventos que permitem com que o usuário possa visualizar, cadastrar, atualizar, cancelar ou consultar mensalidades.

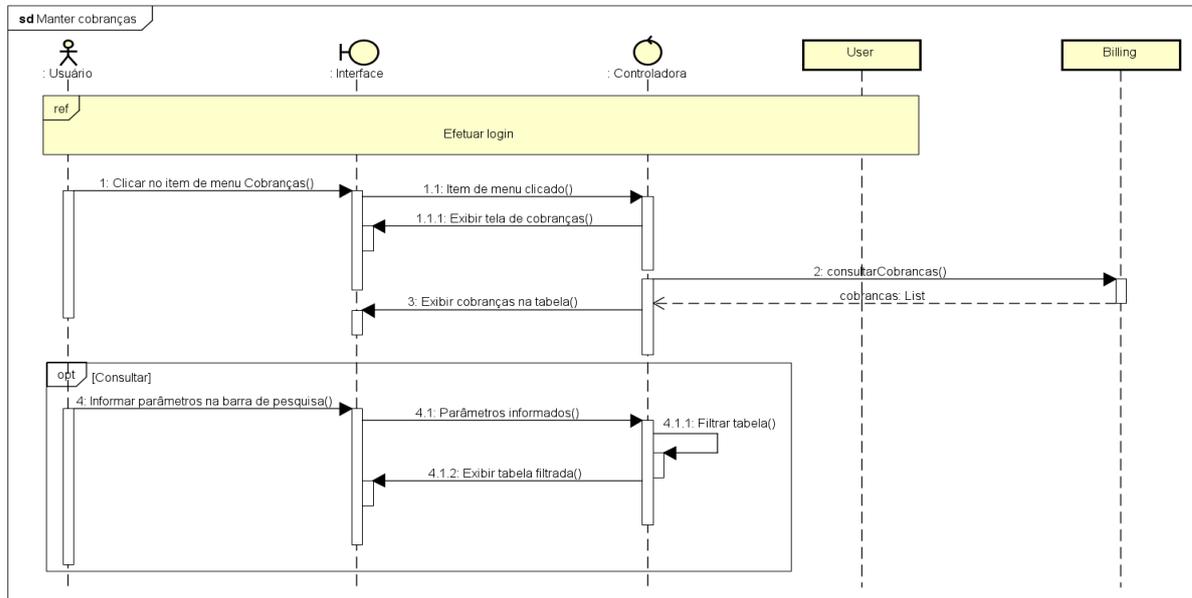
Figura 18 – Diagrama de sequência "Manter mensalidades"



Fonte: Elaboração própria (2022).

A Figura 19 apresenta a sequência de eventos que permitem com que o usuário possa visualizar e consultar as cobranças geradas.

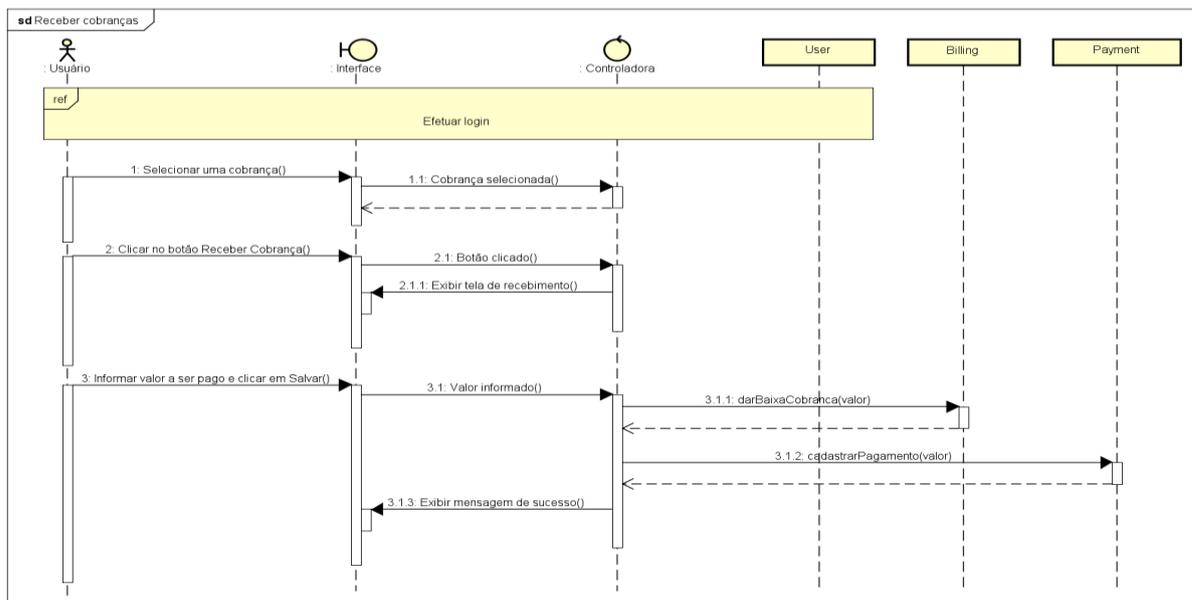
Figura 19 – Diagrama de sequência "Manter cobranças"



Fonte: Elaboração própria (2022).

A Figura 20 apresenta a sequência de eventos que permitem com que o usuário possa receber o pagamento das cobranças.

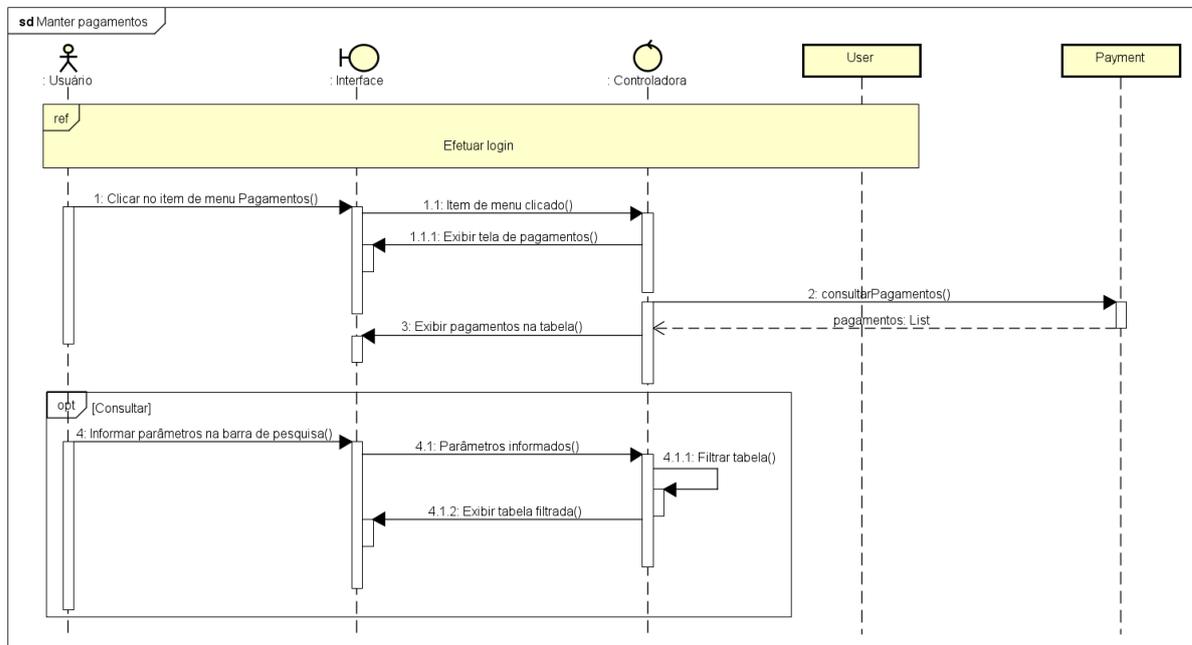
Figura 20 – Diagrama de sequência "Receber cobranças"



Fonte: Elaboração própria (2022).

A Figura 21 apresenta a sequência de eventos que permitem com que o usuário possa visualizar e consultar os pagamentos recebidos.

Figura 21 – Diagrama de sequência "Manter pagamentos"



Fonte: Elaboração própria (2022).

3.4 Banco de dados

Para a Oracle (2014b), uma base de dados ou banco de dados pode ser entendido como um conjunto de dados ou informações estruturadas, geralmente mantidas de maneira organizada em um sistema de computador. Nas palavras de Heuser (2009, p. 22), refere-se a um “conjunto de dados integrados que tem por objetivo atender a uma comunidade de usuários”.

Coronel e Morris (2016, tradução nossa), além de elucidar essa temática de maneira análoga às definições supracitadas – afirmando se tratar de uma estrutura de computador compartilhada e integrada que contém um conjunto de dados do usuário final – expandem essa definição incluindo os metadados, os quais também são armazenados e desempenham um importante papel ao descrever características dos dados e dos relacionamentos que os conectam.

Para lidar com essa estrutura, utiliza-se um sistema de gerenciamento de banco de dados (SGBD), um “software que incorpora as funções de definição, recuperação e alteração de dados em um banco de dados”, e atua como intermediário

(interface) entre o usuário e o banco (HEUSER, 2009, p. 23). Além de mecanismos para a manipulação dos dados, Silberschatz, Korth e Sudarshan (2006) mencionam que o SGBD também precisa garantir a segurança e integridade das informações armazenadas e evitar possíveis falhas, resultados indesejados ou tentativas não autorizadas de acesso.

Atualmente, os dados podem ser considerados um dos ativos mais valiosos para uma empresa, portanto não devem ser tratados de maneira descuidada. Um banco de dados que atende a todos os requisitos do usuário não acontece por acaso. É necessário, antes de tudo, criar um planejamento completo (ou projeto) para compreender os dados e garantir que eles serão usados corretamente.

De acordo com Coronel e Morris (2016, tradução nossa), o projeto de um banco de dados, também conhecido como *design*, consiste em uma etapa que visa estruturá-lo. Para tal, o primeiro passo é realizar a modelagem de dados, que se refere ao processo de criação, com base em um determinado problema, de um modelo de dados específico. Este, por sua vez, é uma representação, tipicamente gráfica, de uma estrutura de dados complexa do mundo real. O modelo de dados final, quando realizado adequadamente, contém todas as instruções para construir um banco de dados que atenderá os requisitos solicitados pelo cliente.

Ainda sobre os modelos de dados, Coronel e Morris (2016, p. 41, tradução nossa) afirmam que:

A busca por um melhor gerenciamento de dados levou a vários modelos que tentam resolver as deficiências críticas do modelo anterior e fornecer soluções para necessidades de gerenciamento de dados em constante evolução. Esses modelos representam escolas de pensamento sobre o que é um banco de dados, o que ele deve fazer, os tipos de estruturas que deve empregar e a tecnologia que seria usada para implementar essas estruturas.

Os conceitos básicos encontrados em todos os modelos são entidades, atributos e relacionamentos. Entidade é uma pessoa, lugar, coisa ou evento sobre o qual os dados serão coletados e armazenados; atributo é uma característica de uma entidade; relacionamento é o conjunto de associações entre entidades, que pode normalmente ocorrer de três maneiras: relacionamento um-para-um (1:1), um-para-muitos (1:n) e muitos-para-muitos (n:n) (HEUSER, 2009).

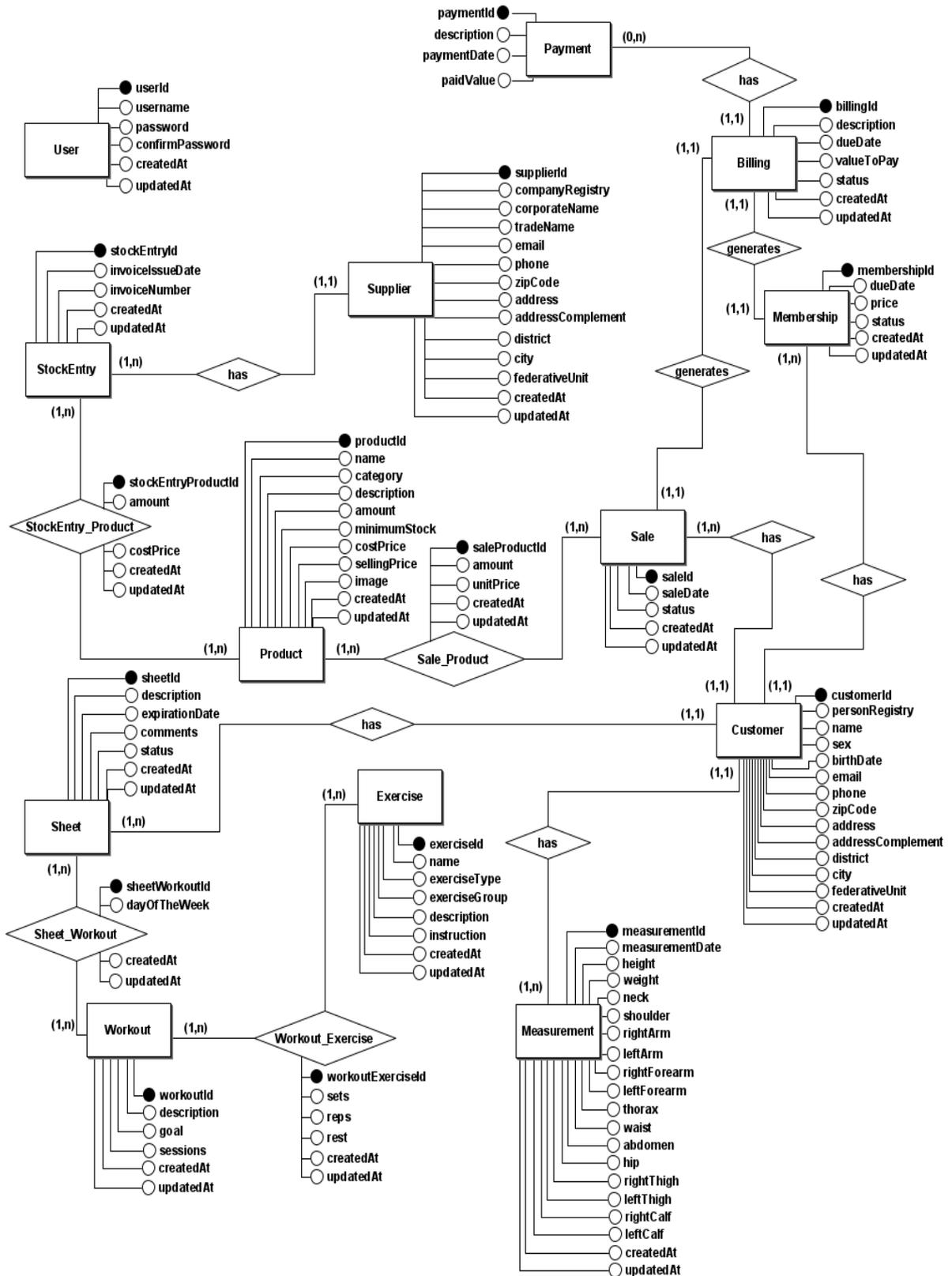
Dentre os modelos existentes, optou-se por utilizar neste projeto o mais difundido, o modelo entidade-relacionamento (modelo ER ou MER). Esse modelo é

geralmente representado graficamente com um diagrama entidade-relacionamento (diagrama ER ou DER), que será o foco do item a seguir.

3.4.1 Diagrama entidade-relacionamento

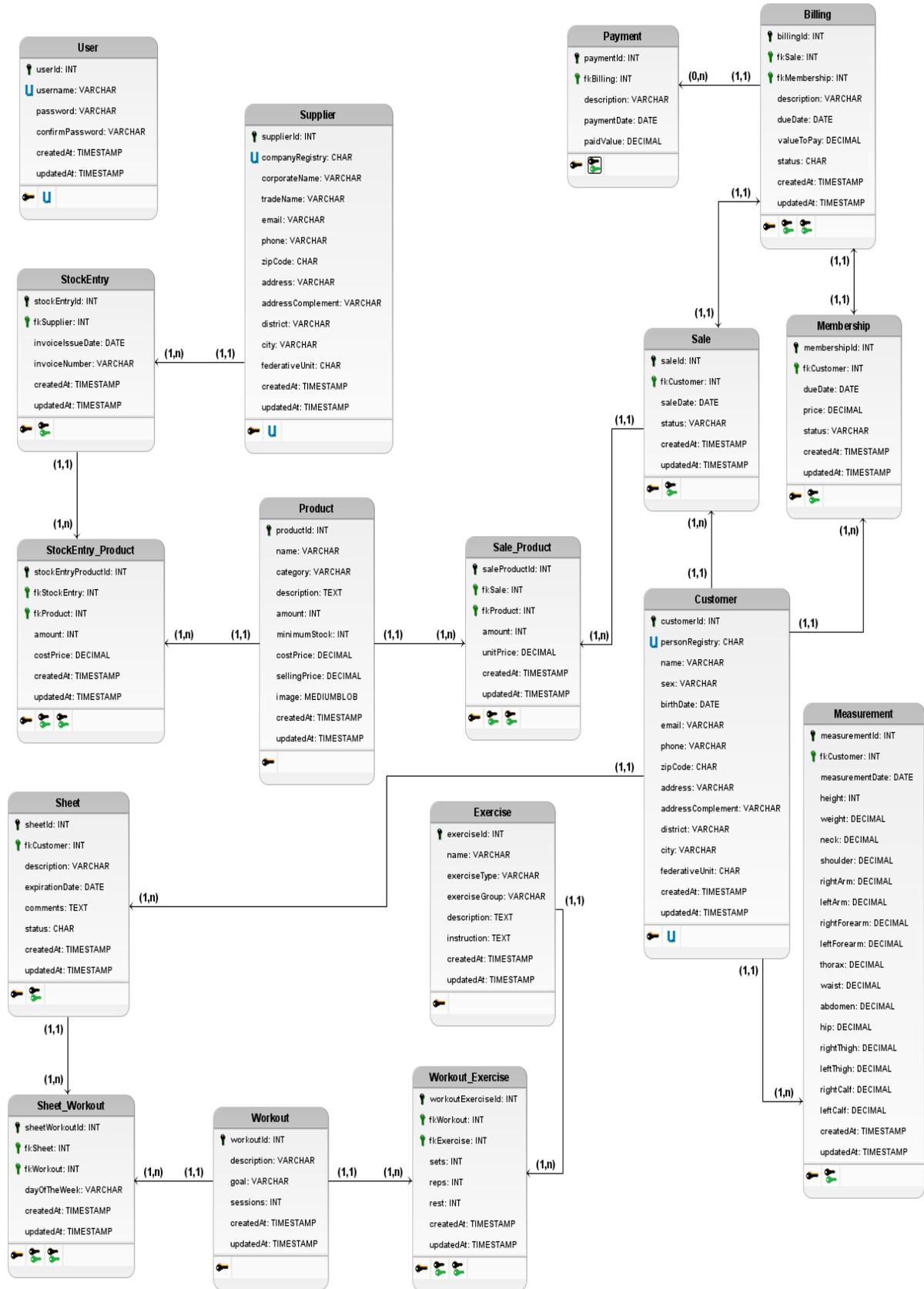
De acordo com Silberschatz, Korth e Sudarshan (2006), diagrama entidade-relacionamento é a representação gráfica das entidades, atributos e relacionamentos de um modelo entidade-relacionamento; são diagramas relativamente simples e inteligíveis utilizados para expressar a estrutura geral de um banco de dados. Sua importância está atrelada ao fato de facilitar a comunicação e promover uma interação eficaz entre as pessoas interessadas no desenvolvimento do projeto. A Figura 22 apresenta o modelo conceitual do banco de dados e a Figura 23, o modelo lógico.

Figura 22 – Modelo conceitual do banco de dados



Fonte: Elaboração própria (2022).

Figura 23 – Modelo lógico do banco de dados



Fonte: Elaboração própria (2022).

3.4.2 Dicionário de dados

O dicionário de dados é um documento que fornece descrições detalhadas das tabelas do banco de dados. Logo, contém definições, significados e restrições sobre os campos que estão sendo armazenados. Em resumo, o dicionário de dados contém metadados – dados sobre dados. Sua importância se deve ao fato de estabelecer convenções ou padrões que são utilizados pelos membros de uma equipe durante um projeto, reduzindo assim as inconsistências e interpretações ambíguas (CORONEL; MORRIS, 2016, tradução nossa). Os Quadros 23 a 39 estão dedicados à apresentação dos dicionários de dados.

Quadro 23 – Dicionário de dados da entidade *User*

Entidade: <i>User</i>					
Descrição: Armazena os usuários do sistema					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>userId</i>	Código identificador do usuário do sistema, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>username</i>	Nome do usuário	Texto	32	Não	UNIQUE
<i>password</i>	Senha do usuário	Texto	64	Não	-
<i>confirmPassword</i>	Confirmação da senha	Texto	64	Não	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 24 – Dicionário de dados da entidade *Supplier*

Entidade: <i>Supplier</i>					
Descrição: Armazena os fornecedores					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>supplierId</i>	Código identificador do fornecedor, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>companyRegistry</i>	Cadastro Nacional da Pessoa Jurídica (CNPJ) do fornecedor	Texto	14	Não	<i>UNIQUE</i>
<i>corporateName</i>	Razão social do fornecedor	Texto	255	Não	-
<i>tradeName</i>	Nome fantasia do fornecedor	Texto	255	Não	-
<i>email</i>	Endereço de <i>e-mail</i> do fornecedor	Texto	255	Sim	-
<i>phone</i>	Telefone fixo ou celular do fornecedor	Texto	11	Sim	-
<i>zipCode</i>	Código de Endereçamento Postal (CEP) da localização do fornecedor	Texto	8	Sim	-
<i>address</i>	Endereço onde o fornecedor se localiza	Texto	255	Sim	-
<i>addressComplement</i>	Complemento do endereço	Texto	255	Sim	-
<i>district</i>	Bairro onde o fornecedor se localiza	Texto	255	Sim	-
<i>city</i>	Cidade onde o fornecedor se localiza	Texto	255	Sim	-
<i>federativeUnit</i>	Unidade federativa da localização do fornecedor	Texto	2	Sim	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 25 – Dicionário de dados da entidade *Product*

Entidade: <i>Product</i>					
Descrição: Armazena os produtos					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>productId</i>	Código identificador do produto, gerado automaticamente	Numérico	-	Não	PRIMARY KEY
<i>name</i>	Nome do produto	Texto	255	Não	-
<i>category</i>	Categoria do produto	Texto	255	Sim	-
<i>description</i>	Detalhes do produto	Texto	-	Sim	-
<i>amount</i>	Quantidade inicial/em estoque do produto	Numérico	-	Não	-
<i>minimumStock</i>	Quantidade mínima do produto	Numérico	-	Não	-
<i>costPrice</i>	Preço de custo do produto	Numérico	(10, 2)	Não	-
<i>sellingPrice</i>	Preço de venda do produto	Numérico	(10, 2)	Não	-
<i>image</i>	Imagem do produto	Binário	-	Sim	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 26 – Dicionário de dados da entidade *StockEntry*

Entidade: <i>StockEntry</i>					
Descrição: Armazena as entradas de estoque					
Observação: Possui chave estrangeira com <i>Supplier</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>stockEntryId</i>	Código identificador da entrada de estoque, gerado automaticamente	Numérico	-	Não	PRIMARY KEY
<i>fkSupplier</i>	Código identificador do fornecedor	Numérico	-	Não	FOREIGN KEY
<i>invoiceIssueDate</i>	Data de emissão da nota fiscal	Data	-	Não	-
<i>invoiceNumber</i>	Número da nota fiscal	Texto	10	Sim	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 27 – Dicionário de dados da entidade *StockEntry_Product*

Entidade: <i>StockEntry_Product</i>					
Descrição: Armazena os produtos inseridos em cada entrada de estoque					
Observação: Possui chave estrangeira com <i>StockEntry</i> e <i>Product</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>stockEntryProductId</i>	Código identificador do item de entrada de estoque, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>fkStockEntry</i>	Código identificador da entrada de estoque	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>fkProduct</i>	Código identificador do produto	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>amount</i>	Quantidade de produto que entrou no estoque	Numérico	-	Não	-
<i>costPrice</i>	Preço de custo de cada produto	Numérico	(10, 2)	Não	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 28 – Dicionário de dados da entidade *Customer*

Entidade: <i>Customer</i>					
Descrição: Armazena os clientes					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>customerId</i>	Código identificador do cliente, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>personRegistry</i>	Cadastro de Pessoa Física (CPF) do cliente	Texto	11	Não	<i>UNIQUE</i>
<i>name</i>	Nome do cliente	Texto	255	Não	-
<i>sex</i>	Sexo do cliente	Texto	9	Não	-
<i>birthDate</i>	Data de nascimento do cliente	Data	-	Não	-
<i>email</i>	Endereço de <i>e-mail</i> do cliente	Texto	255	Sim	-
<i>phone</i>	Telefone fixo ou celular do cliente	Texto	11	Sim	-
<i>zipCode</i>	Código de Endereçamento Postal (CEP) do endereço do cliente	Texto	8	Sim	-
<i>address</i>	Endereço onde o cliente mora	Texto	255	Sim	-
<i>addressComplement</i>	Complemento do endereço	Texto	255	Sim	-
<i>district</i>	Bairro onde o cliente mora	Texto	255	Sim	-
<i>city</i>	Cidade onde o cliente mora	Texto	255	Sim	-
<i>federativeUnit</i>	Unidade federativa onde o cliente mora	Texto	2	Sim	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 29 – Dicionário de dados da entidade *Sale*

Entidade: <i>Sale</i>					
Descrição: Armazena as vendas realizadas					
Observação: Possui chave estrangeira com <i>Customer</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>saleId</i>	Código identificador da venda, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>fkCustomer</i>	Código identificador do cliente	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>saleDate</i>	Data da venda	Data	-	Não	-
<i>status</i>	Situação da venda (paga ou aguardando)	Texto	20	Não	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 30 – Dicionário de dados da entidade *Sale_Product*

Entidade: <i>Sale_Product</i>					
Descrição: Armazena os produtos inseridos em cada venda					
Observação: Possui chave estrangeira com <i>Sale</i> e <i>Product</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>saleProductId</i>	Código identificador do item de venda, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>fkSale</i>	Código identificador da venda	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>fkProduct</i>	Código identificador do produto	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>amount</i>	Quantidade vendida	Numérico	-	Não	-
<i>unitPrice</i>	Preço unitário do produto vendido	Numérico	(10, 2)	Não	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 31 – Dicionário de dados da entidade *Exercise*

Entidade: <i>Exercise</i>					
Descrição: Armazena os exercícios físicos					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>exerciseld</i>	Código identificador do exercício, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>name</i>	Nome do exercício	Texto	255	Não	-
<i>exerciseType</i>	Tipo do exercício	Texto	255	Não	-
<i>exerciseGroup</i>	Grupo de músculos trabalhados pelo exercício	Texto	255	Não	-
<i>description</i>	Descrição do exercício	Texto	-	Sim	-
<i>instruction</i>	Instruções de execução do exercício	Texto	-	Sim	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 32 – Dicionário de dados da entidade *Workout*

Entidade: <i>Workout</i>					
Descrição: Armazena os treinos					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>workoutId</i>	Código identificador do treino, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>description</i>	Descrição do treino	Texto	255	Não	-
<i>goal</i>	Propósito ou objetivo do treino	Texto	255	Não	-
<i>sessions</i>	Quantidade de sessões do treino	Numérico	-	Não	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 33 – Dicionário de dados da entidade *Workout_Exercise*

Entidade: <i>Workout_Exercise</i>					
Descrição: Armazena os exercícios inseridos em cada treino					
Observação: Possui chave estrangeira com <i>Workout</i> e <i>Exercise</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>workoutExerciseId</i>	Código identificador do item de treino, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>fkWorkout</i>	Código identificador do treino	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>fkExercise</i>	Código identificador do exercício	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>sets</i>	Quantidade de séries do exercício que devem ser feitas	Numérico	-	Não	-
<i>reps</i>	Quantidade de repetições do exercício que devem ser feitas	Numérico	-	Não	-
<i>rest</i>	Tempo de descanso (em segundos) entre as séries do exercício	Numérico	-	Não	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 34 – Dicionário de dados da entidade *Measurement*

Entidade: <i>Measurement</i>					
Descrição: Armazena as medidas dos perímetros corporais de clientes					
Observação: Possui chave estrangeira com <i>Customer</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>measurementId</i>	Código identificador da medida, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>fkCustomer</i>	Código identificador do cliente	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>measurementDate</i>	Data em que a medição foi feita	Data	-	Não	-
<i>height</i>	Altura (em cm) do cliente	Numérico	-	Não	-
<i>weight</i>	Peso (em kg) do cliente	Numérico	(4, 1)	Não	-
<i>neck</i>	Medida do pescoço (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>shoulder</i>	Medida dos ombros (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>rightArm</i>	Medida do braço direito (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>leftArm</i>	Medida do braço esquerdo (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>rightForearm</i>	Medida do antebraço direito (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>leftForearm</i>	Medida do antebraço esquerdo (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>thorax</i>	Medida do tórax (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>waist</i>	Medida da cintura (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>abdomen</i>	Medida do abdômen (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>hip</i>	Medida do quadril (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>rightThigh</i>	Medida da coxa direita (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>leftThigh</i>	Medida da coxa esquerda (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>rightCalf</i>	Medida da panturrilha direita (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>leftCalf</i>	Medida da panturrilha esquerda (em cm) do cliente	Numérico	(4, 1)	Sim	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 35 – Dicionário de dados da entidade *Sheet*

Entidade: <i>Sheet</i>					
Descrição: Armazena as fichas de treino					
Observação: Possui chave estrangeira com <i>Customer</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>sheetId</i>	Código identificador da ficha de treino, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>fkCustomer</i>	Código identificador do cliente	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>description</i>	Descrição da ficha de treino	Texto	255	Não	-
<i>expirationDate</i>	Data de validade da ficha de treino	Data	-	Não	-
<i>comments</i>	Observações da ficha de treino	Texto	-	Sim	
<i>status</i>	Situação atual da ficha de treino	Texto	1	Não	
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 36 – Dicionário de dados da entidade *Sheet_Workout*

Entidade: <i>Sheet_Workout</i>					
Descrição: Armazena os treinos inseridos em cada ficha					
Observação: Possui chave estrangeira com <i>Sheet</i> e <i>Workout</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>sheetWorkoutId</i>	Código identificador do item de ficha, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>fkSheet</i>	Código identificador da ficha	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>fkWorkout</i>	Código identificador do treino	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>dayOfTheWeek</i>	Dia da semana em que o treino irá ocorrer	Texto	13	Não	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 37 – Dicionário de dados da entidade *Membership*

Entidade: <i>Membership</i>					
Descrição: Armazena as mensalidades dos clientes					
Observação: Possui chave estrangeira com <i>Customer</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>membershipId</i>	Código identificador da mensalidade, gerado automaticamente	Numérico	-	Não	PRIMARY KEY
<i>fkCustomer</i>	Código identificador do cliente	Numérico	-	Não	FOREIGN KEY
<i>dueDate</i>	Data de vencimento da mensalidade	Data	-	Não	-
<i>price</i>	Preço da mensalidade	Numérico	(10, 2)	Não	-
<i>status</i>	Situação da mensalidade (paga ou aguardando)	Texto	20	Não	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 38 – Dicionário de dados da entidade *Billing*

Entidade: <i>Billing</i>					
Descrição: Armazena as cobranças geradas a partir das vendas e mensalidades					
Observação: Possui chave estrangeira opcional com <i>Sale</i> e <i>Membership</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>billingId</i>	Código identificador da cobrança, gerado automaticamente	Numérico	-	Não	PRIMARY KEY
<i>fkSale</i>	Código identificador da venda	Numérico	-	Sim	FOREIGN KEY
<i>fkMembership</i>	Código identificador da mensalidade	Numérico	-	Sim	FOREIGN KEY
<i>description</i>	Descrição da cobrança	Texto	255	Não	-
<i>dueDate</i>	Data de vencimento da cobrança	Data	-	Não	-
<i>valueToPay</i>	Valor a pagar da cobrança	Numérico	(10, 2)	Não	-
<i>status</i>	Situação da cobrança (ativa ou inativa)	Texto	1	Não	-
<i>createdAt</i>	Data e hora de registro, geradas automaticamente	Data e hora	-	Não	-
<i>updatedAt</i>	Data e hora da última atualização, geradas automaticamente	Data e hora	-	Não	-

Fonte: Elaboração própria (2022).

Quadro 39 – Dicionário de dados da entidade *Payment*

Entidade: <i>Payment</i>					
Descrição: Armazena os pagamentos das cobranças geradas					
Observação: Possui chave estrangeira com <i>Billing</i>					
Atributo	Descrição	Domínio	Tamanho (caracteres)	Nulo	Índice
<i>paymentId</i>	Código identificador do pagamento, gerado automaticamente	Numérico	-	Não	<i>PRIMARY KEY</i>
<i>fkBilling</i>	Código identificador da cobrança	Numérico	-	Não	<i>FOREIGN KEY</i>
<i>description</i>	Descrição do pagamento	Texto	255	Não	-
<i>paymentDate</i>	Data de pagamento da cobrança	Data	-	Não	-
<i>paidValue</i>	Valor pago da cobrança	Numérico	(10, 2)	Não	-

Fonte: Elaboração própria (2022).

4 PROJETO

Este capítulo irá detalhar o funcionamento da metodologia empregada neste trabalho, as etapas de desenvolvimento e as interfaces desenvolvidas usando o JavaFX, capturadas a partir do sistema em plena execução.

4.1 Metodologia

As metodologias ágeis são abordagens que estão alinhadas com os valores e princípios descritos no *Manifesto Ágil*. Pensado e elaborado por 17 profissionais da área de desenvolvimento de *software*, o documento foi publicado em 2001 como uma alternativa às metodologias tradicionais utilizadas naquela época, que apresentavam processos longos, objetivos rígidos e sequenciais e muita documentação. Por outro lado, as metodologias ágeis contam com uma proposta de processos mais curtos, com entregas em um espaço de tempo menor, e permitem que as equipes se adaptem rapidamente às mudanças, pois reconhecem que as necessidades do cliente não são estáticas (AGILE ALLIANCE, 2019, tradução nossa).

Durante a realização deste projeto, optou-se por utilizar uma das metodologias de desenvolvimento ágil mais populares atualmente, o *Scrum*. Criado por Ken Schwaber e Jeff Sutherland, trata-se de uma estrutura para desenvolver, entregar e sustentar produtos em ambientes complexos, dando ênfase nas pessoas da equipe (SCHWABER; SUTHERLAND, 2020). O *Scrum* é baseado em um processo iterativo e incremental, e como relembra Atlassian (2022), no aprendizado contínuo, pois reconhece que a equipe não sabe tudo no início de um projeto e que evoluirá com a experiência. Outras características citadas por Schwaber (1997) incluem: entrega e cronograma flexíveis, equipes pequenas, frequentes revisões e colaboração.

O nome dessa metodologia surgiu a partir da comparação com o jogo de *rugby*, pois *Scrum* é uma reunião que acontece quando os jogadores de *rugby* vão iniciar um lance, sendo essa a inspiração para a criação do nome. A analogia foi utilizada visto que os times desse esporte agem em conformidade, assim cada membro tem um papel específico dentro da determinada estratégia e todos se ajudam para alcançar um objetivo comum (é assim que os times de desenvolvimento que utilizam o método *Scrum* devem se comportar) (SCHWABER, 1997).

Segundo Schwaber e Sutherland (2020), o *Scrum* começa com o *Product Backlog*, uma lista emergente e ordenada por nível de prioridade de conclusão de tudo que é necessário alcançar (características, funções e requisitos) para desenvolver ou aprimorar um produto de *software*. Essa lista é feita em uma reunião com todas as pessoas envolvidas no projeto, por exemplo, desenvolvedores, investidores, clientes e parceiros.

O desenvolvimento do projeto é dividido em ciclos pequenos de tempo chamados de *Sprints* – a parte principal do *Scrum*. Schwaber e Sutherland (2020, p. 8) conceituam:

A *Sprint* é um contêiner para todos os outros eventos. [...] *Sprints* são o coração do *Scrum*, onde ideias são transformadas em valor. São eventos de duração fixa de um mês ou menos para criar consistência. Uma nova *Sprint* começa imediatamente após a conclusão da *Sprint* anterior.

Cada *Sprint* tem a definição do que é para ser construído: um plano projetado e flexível que irá guiar a construção, o trabalho e o resultado do produto. No final, uma versão incremental potencialmente utilizável do sistema é criada.

A partir desse ponto, torna-se necessário descrever brevemente a unidade fundamental do *Scrum*, o *Scrum Team*. Schwaber e Sutherland (2020, p. 6-7) esclarecem que se trata de um pequeno time de pessoas composto por:

- a) *Developers*: “são as pessoas do *Scrum Team* que estão comprometidas em criar qualquer aspecto de um incremento utilizável a cada *Sprint*”; isto é, estão ligados às tarefas mais técnicas;
- b) *Product Owner*: “é responsável por maximizar o valor do produto resultante do trabalho do *Scrum Team*. [...] também é responsável pelo gerenciamento eficaz do *Product Backlog* [...]”;
- c) *Scrum Master*: “é responsável por estabelecer o *Scrum* conforme definido no *Guia do Scrum*. [...] é responsável pela eficácia do *Scrum Team*”.

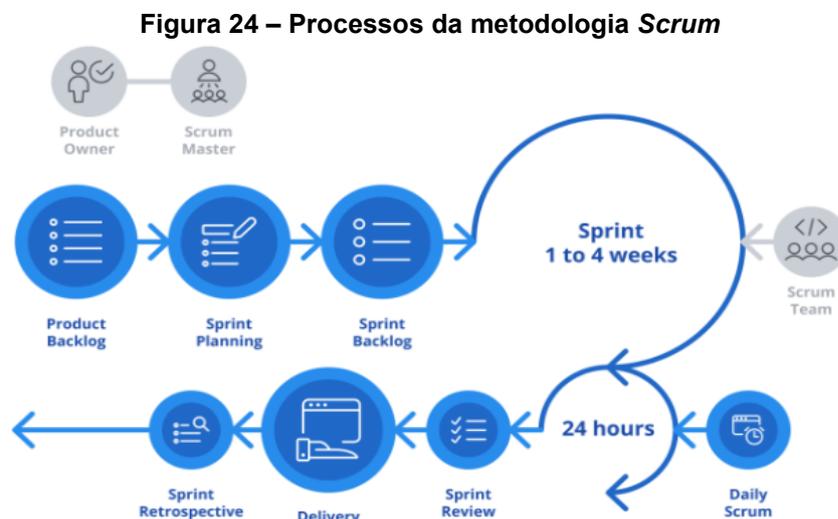
Ainda de acordo com os autores referidos acima, a metodologia *Scrum* determina uma reunião de planejamento no início de cada *Sprint* (*Sprint Planning*), na qual os membros do time definem, com base no *Product Backlog*, o que pode ser entregue e como esse trabalho será realizado. Como resultado desse evento, obtém-se o *Sprint Backlog*, um subconjunto de itens do *Product Backlog* selecionados para

a *Sprint* atual. Durante o andamento desse ciclo, realiza-se diariamente um evento chamado *Daily Scrum*. Em outras palavras, trata-se de uma reunião curta (de até 15 minutos) que ocorre todos os dias entre os membros da equipe do projeto para sincronizar as atividades e criar um plano para as próximas 24 horas. Nessa reunião, para Rising e Janoff (2000, p. 31, tradução nossa), três perguntas devem ser respondidas por cada participante:

O que você completou, em relação ao *Backlog*, desde a última reunião do *Scrum*? Que obstáculos surgiram no seu caminho para concluir este trabalho? Que coisas específicas você planeja realizar, em relação ao *Backlog*, entre agora e a próxima reunião do *Scrum*?

À medida em que tais perguntas vão sendo respondidas pelos integrantes, a equipe toma ciência do nível de comprometimento de cada um com o projeto e consegue organizar as ideias e atividades restantes.

Ao final da *Sprint*, Schwaber e Sutherland (2020) defendem uma revisão (*Sprint Review*), na qual o time apresenta os resultados dos trabalhos para os principais envolvidos do projeto, e uma retrospectiva (*Sprint Retrospective*), cuja intenção é refletir e planejar maneiras de aumentar a qualidade e eficácia dos ciclos futuros. Nessa última etapa, observam-se quais aspectos foram concluídos com sucesso, quais foram os problemas encontrados durante o período de desenvolvimento e como esses problemas foram (ou não) solucionados, e cria-se um plano com as ações de mudanças mais impactantes para a próxima *Sprint*. A Figura 24 ilustra, de maneira concisa, as etapas da metodologia *Scrum* descritas acima.



Fonte: Sydle (2020).

4.2 Etapas de desenvolvimento

Com base na metodologia *Scrum* apresentada no item 4.1, esta seção irá descrever brevemente os acontecimentos de cada *Sprint* (ciclo de desenvolvimento) do projeto. Por questões metodológicas, ficou definido que os ciclos teriam a duração de 14 dias (a única exceção foi a *Sprint* 3, que teve uma duração de 21 dias). No início de cada *Sprint*, a equipe se reunia para definir as atividades a serem desenvolvidas e ao final, apresentava os resultados e realizava uma retrospectiva para dialogar e debater a respeito dos aspectos que deram certo e dos aspectos que deram errado, e identificar ações de melhorias para aplicar nos futuros ciclos.

4.2.1 *Sprint* 1

No dia 14 de março de 2022, a equipe se reuniu com o objetivo de planejar a primeira entrega seguindo a metodologia *Scrum*, a qual foi nomeada como *Sprint* 1, com data-limite marcada para o dia 27 de março.

Durante a reunião, ficou definido entre os integrantes a execução de atividades relativamente mais básicas para o início do projeto, garantindo assim tempo suficiente para estudar os recursos e ferramentas (que até então não eram muito familiares ou utilizados com frequência) e aplicar os conhecimentos adquiridos de maneira apropriada.

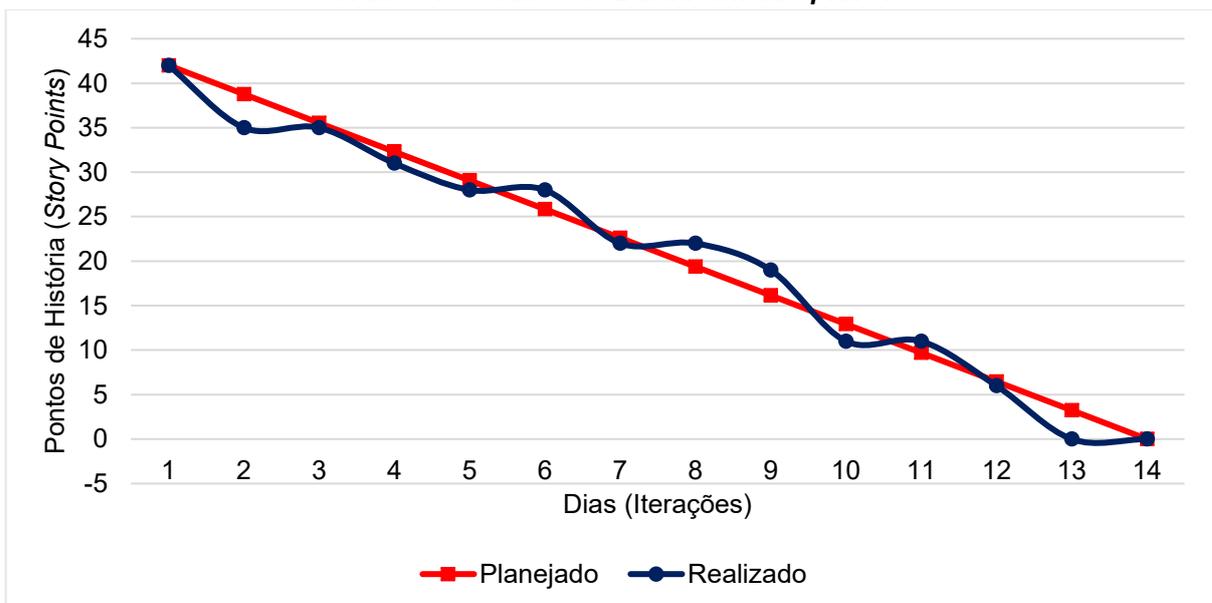
De modo geral, as atividades propostas para a *Sprint* 1 estavam relacionadas com o cadastro de usuário, autenticação, acesso à tela principal do sistema e menu de opções. Com a intenção de medir o esforço requerido ou dificuldade para implementá-las por completo, cada uma recebeu um certo número de pontos de história (quanto maior a pontuação, maior a dificuldade ou tempo necessário para concluir). O Quadro 40 apresenta as atividades realizadas e suas respectivas pontuações.

Quadro 40 – Atividades realizadas durante a *Sprint 1*

Atividades	Pontos
Criar a classe modelo para usuários	1
Criar o banco de dados SQL	2
Criar uma tabela SQL para armazenar os usuários	2
Estabelecer uma conexão com o banco de dados SQL utilizando a API JDBC	2
Implementar o padrão DAO em Java para este modelo	2
Incluir as dependências .JAR no diretório de bibliotecas do sistema	2
Criar a tela de cadastro de usuário	3
Criar a tela de <i>login</i>	3
Validar os campos	3
Desenvolver os métodos de cadastro e autenticação	3
Criar a tela principal	3
Criar as telas de <i>mockup</i>	5
Programar a navegação entre as telas de <i>mockup</i>	5
Definir atalhos de teclado para os itens de menu da tela principal	2
Criar o botão de <i>logout</i>	1
Realizar testes com o intuito de identificar problemas	3
Total	42

Fonte: Elaboração própria (2022).

Nesse período, a equipe foi efetuando baixas nas atividades à medida em que elas eram concluídas, obtendo, ao final, um gráfico de *Burndown*, visto no Gráfico 1.

Gráfico 1 – Gráfico de *Burndown* da *Sprint 1*

Fonte: Elaboração própria (2022).

De acordo com Schwaber (2004), um dos criadores da metodologia *Scrum*, o gráfico de *Burndown* é uma representação visual do trabalho restante de um projeto (ou *Sprint*) ao longo do tempo. Em outras palavras, ele afirma que “é a colisão da realidade (trabalho feito e quão rápido está sendo feito) com o que é planejado ou esperado”.

Essa representação gráfica é formada por dois eixos: Y (vertical), que representa os pontos de história, isto é, as atividades pendentes, e o X (horizontal), que corresponde ao tempo decorrido em dias úteis. O ponto mais alto (à esquerda do gráfico) indica o início da *Sprint*, e o ponto mais baixo (à direita do gráfico), o final da *Sprint*.

A linha reta vermelha que conecta esses dois pontos atua como um guia para a equipe e pode ser chamada de linha de trabalho ideal, criada a partir da soma das estimativas de todas as tarefas que precisam ser concluídas. Por outro lado, a linha azul é a linha de trabalho real.

Dependendo da quantidade de esforços empregados pelos integrantes de uma equipe e à medida em que o tempo avança, a linha real irá flutuar acima ou abaixo da ideal, indicando assim se o projeto está atrasado ou adiantado respectivamente. A partir dessa explicação e observando o gráfico, percebe-se que tudo ocorreu bem ao longo do desenvolvimento da primeira entrega, conforme o planejamento.

Ao final da *Sprint*, no dia 26 de março de 2022, a equipe se reuniu novamente, mas dessa vez com a intenção de realizar uma retrospectiva, ou seja, dialogar e debater a respeito dos aspectos que deram certo e dos aspectos que deram errado, e identificar ações de melhorias para as futuras entregas, sempre com foco no aperfeiçoamento. As conclusões obtidas encontram-se a seguir:

- a) O que deu certo: as atividades foram concluídas conforme o planejado, surgiram boas ideias para as entregas futuras e muitos conhecimentos sobre as ferramentas foram adquiridos nesse período;
- b) O que deu errado: comunicação e proatividade inconstantes e perda de tempo com detalhes não essenciais (tempo esse que poderia ser realocado em outras atividades);
- c) Ações de melhorias: estabelecer mais reuniões curtas (presenciais ou remotas) ao longo da semana, deixando assim todos os integrantes cientes da situação do projeto, e desenvolver um pouco da documentação ou do sistema todos os dias.

4.2.2 *Sprint 2*

No dia 28 de março de 2022, os integrantes se reuniram de novo para começar a preparar a segunda entrega seguindo os processos da metodologia *Scrum*, dessa vez com a data de finalização marcada para o dia 10 de abril.

A equipe definiu que todas as atividades dessa *Sprint* estariam relacionadas com a parte de fornecedores da academia, por esse motivo era necessário implementar as quatro operações básicas realizadas em um banco de dados, o CRUD (*Create, Read, Update, Delete*), ou em português, respectivamente, criar, ler, atualizar e excluir registros, descritas brevemente a seguir:

- a) *create*: consiste em uma função que é chamada quando um novo fornecedor está sendo adicionado;
- b) *read*: refere-se a uma função para visualizar todos os fornecedores cadastrados no sistema; essa chamada simplesmente recupera os registros e exibe os resultados;
- c) *update*: responsável por efetuar a alteração das informações de um determinado fornecedor;
- d) *delete*: deve haver, por fim, uma função a ser chamada para remover um fornecedor da lista.

Para medir o esforço requerido ou dificuldade de cada atividade foi utilizado o sistema de pontos de história novamente. O Quadro 41 apresenta cada uma das atividades realizadas nessa *Sprint* e sua respectiva pontuação.

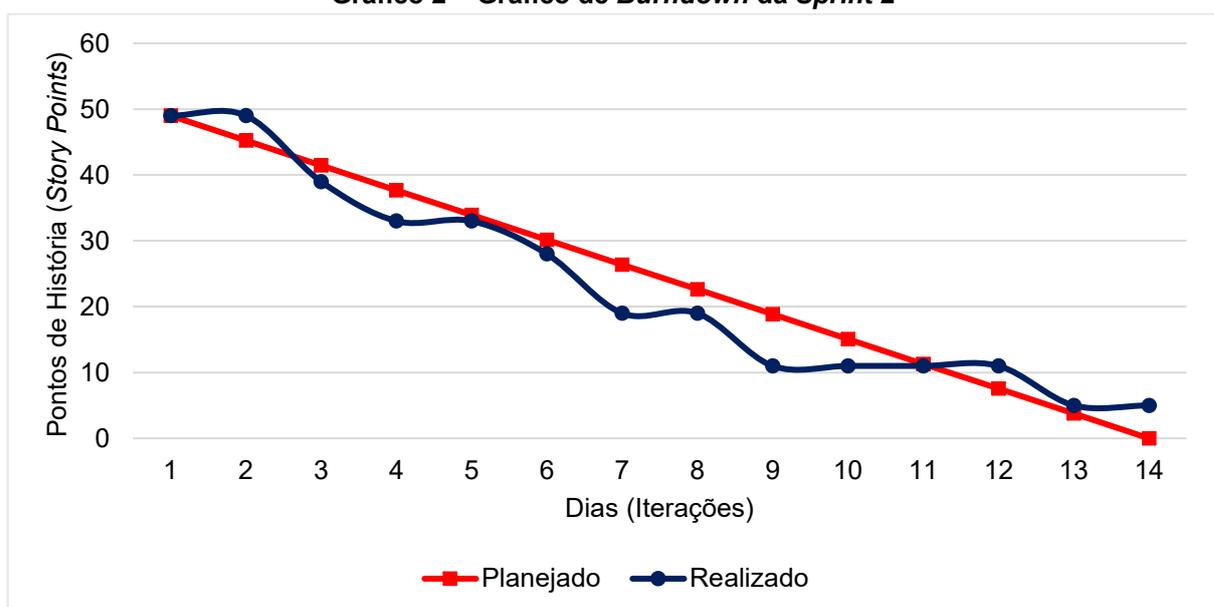
Quadro 41 – Atividades realizadas durante a *Sprint 2*

Atividades	Pontos
Criar a classe modelo para fornecedores	1
Criar uma tabela SQL para armazenar os fornecedores	2
Implementar o padrão DAO em Java para este modelo	2
Incluir mais dependências .JAR no diretório de bibliotecas do sistema	2
Criar a tela principal de fornecedores	3
Criar a tela de cadastro de fornecedor	3
Criar um componente de texto do JavaFX personalizado	5
Criar um validador de CNPJ	5
Criar a tela de atualização de fornecedor	3
Validar o restante dos campos	2
Desenvolver os métodos de cadastro e atualização	2
Exibir os principais atributos em uma tabela de visualização do JavaFX	5
Exibir todos os atributos na seção de detalhes da tela principal	3
Criar o botão de excluir fornecedor	1
Programar a exibição de alertas para o usuário por meio de caixas de diálogo	2
Criar a barra de pesquisa de fornecedor e os botões de filtro	5
Realizar testes com o intuito de identificar problemas	3
Total	49

Fonte: Elaboração própria (2022).

Durante o período de desenvolvimento, baixas foram sendo feitas nas atividades à medida em que elas eram concluídas, obtendo o Gráfico 2.

Gráfico 2 – Gráfico de *Burndown* da *Sprint 2*



Fonte: Elaboração própria (2022).

Dessa vez, a partir da observação da linha de trabalho real (linha azul), percebe-se que não foi possível concluir tudo que foi planejado. Por se tratar de um período do semestre com muitas avaliações e trabalhos, uma das atividades não pôde ser completamente desenvolvida, a barra de pesquisa de fornecedor.

Muitas horas foram dedicadas para estudar um modo de implementar tal funcionalidade, e a equipe até que conseguiu algum progresso, porém ainda não estava plenamente funcional. Ao inserir o valor para buscar na barra de pesquisa, a tabela responsável por apresentar os fornecedores cadastrados deveria ser filtrada, exibindo somente os dados correspondentes ao valor digitado, entretanto, em diversos momentos, isso não acontecia dessa maneira. Em consequência desse mau funcionamento e da ausência de uma solução, a barra de pesquisa precisou ser adiada para a próxima *Sprint*.

Sobre a retrospectiva, o resumo de alguns tópicos discutidos encontra-se a seguir:

- a) O que deu certo: as operações CRUD foram desenvolvidas com sucesso (podem ser reaproveitadas em outras ocasiões do projeto, basta adaptá-las);
- b) O que deu errado: o desenvolvimento da barra de pesquisa de fornecedor.
- c) Ações de melhorias: migrar o projeto para uma versão mais recente do Java para ter acesso a mais recursos.

4.2.3 *Sprint 3*

Diferentemente das demais *Sprints* até aqui, a *Sprint 3* teve um período maior para a sua conclusão por causa da semana de recesso acadêmico. A terceira entrega conforme os processos da metodologia *Scrum* teve início em 11 de abril de 2022, com data-limite marcada para o dia 1 de maio.

Durante uma reunião, ficou definido entre os integrantes que as atividades dessa *Sprint* estariam relacionadas, em sua maioria, com a parte dos produtos oferecidos pela academia. Dessa forma, seria necessário implementar novamente as quatro operações básicas realizadas em um banco de dados, o CRUD.

Antes de começar a desenvolvê-las efetivamente, a equipe teve que dedicar um certo tempo para resolver uma pendência da *Sprint 2*. Como descrito no item 4.2.2,

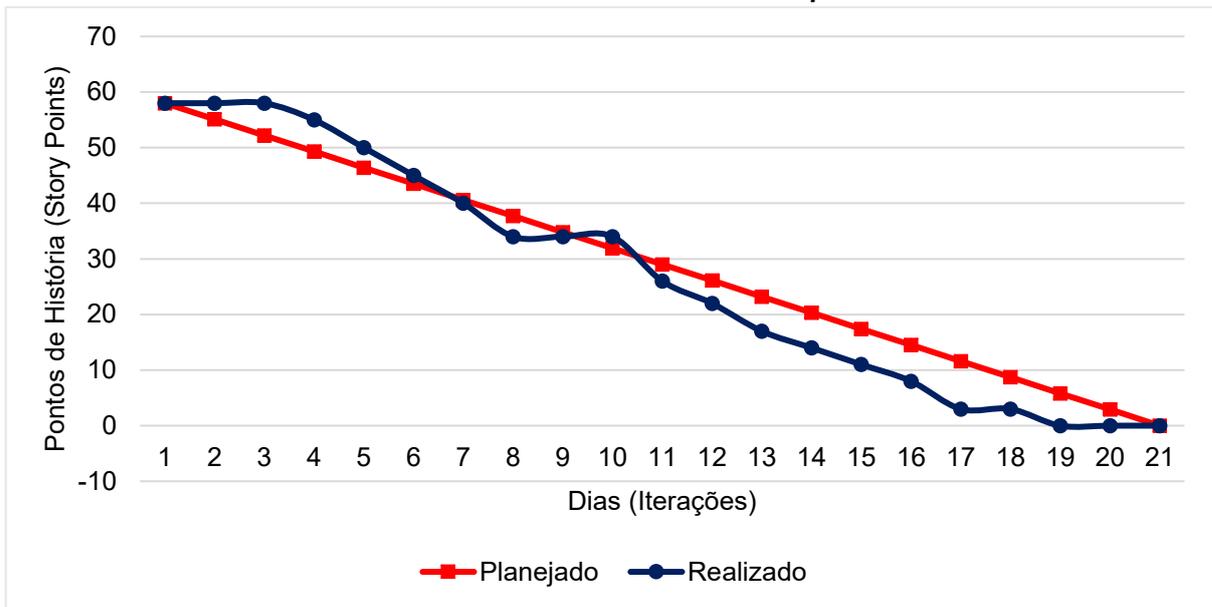
a barra de pesquisa de fornecedores não pôde ser concluída dentro do prazo, já que não estava funcionando corretamente, então precisou ser adiada e incluída nesta entrega. No fim de seu desenvolvimento – que demandou um esforço considerável – pôde ser reaproveitada no módulo de produtos. Essa e as demais atividades da *Sprint* 3 podem ser visualizadas no Quadro 42, assim como os pontos de história de cada uma.

Quadro 42 – Atividades realizadas durante a *Sprint* 3

Atividades	Pontos
Corrigir os erros que foram identificados em atividades da <i>Sprint</i> 2	5
Reformular a barra de pesquisa de fornecedor	5
Criar a classe modelo para produtos	1
Criar uma tabela SQL para armazenar os produtos	2
Implementar o padrão DAO em Java para este modelo	2
Criar a tela principal de produtos	3
Criar a tela de cadastro de produto	3
Manipular imagens utilizando o Java e o banco de dados SQL	8
Criar a tela de atualização de produto	3
Validar os campos	2
Desenvolver os métodos de cadastro e atualização	2
Exibir os principais atributos em uma tabela de visualização do JavaFX	5
Criar métodos para aperfeiçoar a visualização dos dados na tabela	3
Exibir todos os atributos na seção de detalhes da tela principal	3
Criar o botão de excluir produto	1
Programar a exibição de alertas para o usuário por meio de caixas de diálogo	2
Criar a barra de pesquisa de produto e os botões de filtro	5
Realizar testes com o intuito de identificar problemas	3
Total	58

Fonte: Elaboração própria (2022).

Durante o período de desenvolvimento, a equipe foi efetuando baixas nas atividades à medida em que elas eram concluídas, obtendo, ao final, outro gráfico de *Burndown*, conforme o Gráfico 3. Ao interpretá-lo, percebe-se que o começo da terceira *Sprint* foi um pouco conturbado, visto que a linha de trabalho real ficou muitos dias acima da linha de trabalho ideal. Apesar disso, foi possível contornar todas as adversidades e recuperar esse tempo de modo competente a partir da segunda metade do eixo X, em que é possível observar que as atividades planejadas foram finalizadas alguns dias antes do prazo definido (linha de trabalho real abaixo da linha de trabalho ideal).

Gráfico 3 – Gráfico de *Burndown* da *Sprint 3*

Fonte: Elaboração própria (2022).

No fim do período de desenvolvimento, um dia antes da data-limite, foi realizada uma retrospectiva da *Sprint 3*, obtendo algumas conclusões, a saber:

- a) O que deu certo: como citado na retrospectiva da *Sprint 2* no item 4.2.2, a ação de melhoria proposta para a *Sprint 3* estava relacionada com a migração do projeto para uma versão mais recente do Java, com a justificativa de ter acesso a mais recursos. Nesse caso, esse foi um dos aspectos positivos. O projeto, que antes estava sendo desenvolvido na versão 8 do Java, passou a ser desenvolvido na versão 11. Além disso, a barra de pesquisa de fornecedor foi finalizada e houve uma boa comunicação, de forma a deixar todos cientes da situação atual;
- b) O que deu errado: apesar de algumas dificuldades para implementar alguns métodos, de modo geral, tudo ocorreu bem no final;
- c) Ações de melhorias: cada integrante deve organizar e distribuir melhor o seu tempo disponível.

4.2.4 *Sprint 4*

Em 2 de maio de 2022, a equipe definiu as atividades que seriam realizadas durante a quarta etapa e estabeleceu o prazo para o dia 15 de maio. Uma vez que os módulos de fornecedores e de produtos foram finalizados, tornou-se possível iniciar o

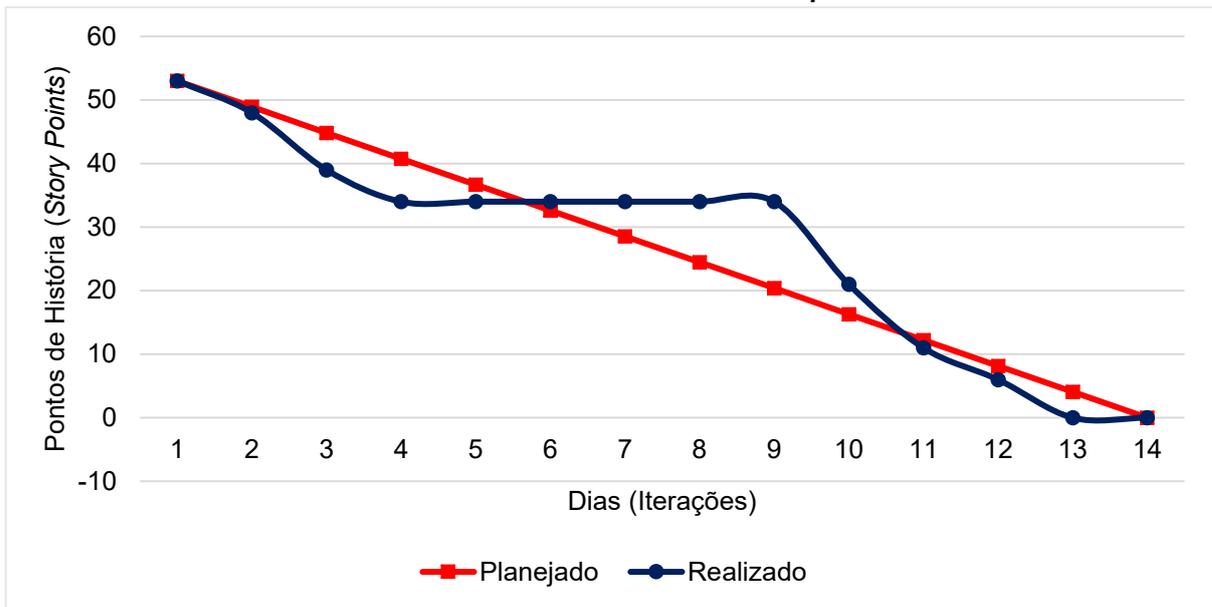
desenvolvimento das funcionalidades do sistema relacionadas às entradas de estoque, visto o seu relacionamento com essas entidades, conforme o diagrama entidade-relacionamento do item 3.4.1. Especificam-se as atividades propostas e pontos de história atribuídos em função do nível de dificuldade enfrentado conforme o Quadro 43.

Quadro 43 – Atividades realizadas durante a *Sprint 4*

Atividades	Pontos
Corrigir os erros que foram identificados em atividades da <i>Sprint 3</i>	5
Criar a classe modelo para entradas de estoque	1
Criar a classe modelo para itens de estoque	1
Criar uma tabela SQL para armazenar as entradas de estoque	2
Criar uma tabela SQL para armazenar os itens de estoque	2
Criar <i>triggers</i> SQL para atualizar a quantidade do produto ao entrar no estoque	1
Implementar o padrão DAO em Java para estes modelos	2
Criar a tela principal de entradas de estoque	5
Criar a tela para inserir itens de estoque	13
Validar os campos e cadastrar	2
Exibir os principais atributos em uma tabela de visualização do JavaFX	5
Exibir todos os atributos na seção de detalhes da tela principal	3
Criar o botão de cancelar entrada de estoque	3
Programar a exibição de alertas para o usuário por meio de caixas de diálogo	2
Criar a barra de pesquisa de entrada de estoque e os botões de filtro	3
Realizar testes com o intuito de identificar problemas	3
Total	53

Fonte: Elaboração própria (2022).

No decorrer das duas semanas de desenvolvimento, a equipe realizou baixas na pontuação no momento em que as atividades eram concluídas. Desse modo, obteve-se outro gráfico de *Burndown*, conforme o Gráfico 4. Ao observá-lo, percebe-se que houve um período de atraso na metade da *Sprint*. Isso ocorreu por causa da atividade referente à criação da tela para inserir os itens de estoque, que demandou conhecimentos que até então não haviam sido aplicados. No entanto, com muita pesquisa, foi possível contornar esse obstáculo e retomar o ritmo normal do projeto nos dias seguintes.

Gráfico 4 – Gráfico de *Burndown* da *Sprint* 4

Fonte: Elaboração própria (2022).

Ao final, houve outra reunião com o intuito de fazer uma retrospectiva desse ciclo e analisar alguns aspectos e propor melhorias para corrigir os problemas identificados. As questões a seguir foram consideradas:

- a) O que deu certo: apesar de ter sido a *Sprint* mais difícil até o momento, justamente por fazer uso de conceitos que ainda não tinham sido praticados, todas as atividades foram concluídas com sucesso;
- b) O que deu errado: desentendimentos e ações que tornavam inviável a continuação do projeto;
- c) Ações de melhorias: cada integrante da equipe deve refletir sobre seu papel na equipe e o quanto está contribuindo de fato para o sucesso do projeto; todos devem ser honestos sobre o que estão pensando.

4.2.5 *Sprint* 5

Em 16 de maio de 2022 se iniciou a quinta etapa de desenvolvimento do projeto, com data-limite marcada para o dia 29 de maio. Depois de aplicar as quatro operações básicas presentes em um banco de dados nos módulos de fornecedores e produtos, os autores deste trabalho concordaram que seria oportuno adiantar o desenvolvimento das funções CRUD relacionadas aos clientes da academia, responsáveis por compor um importante módulo. Além disso, outra atividade relevante

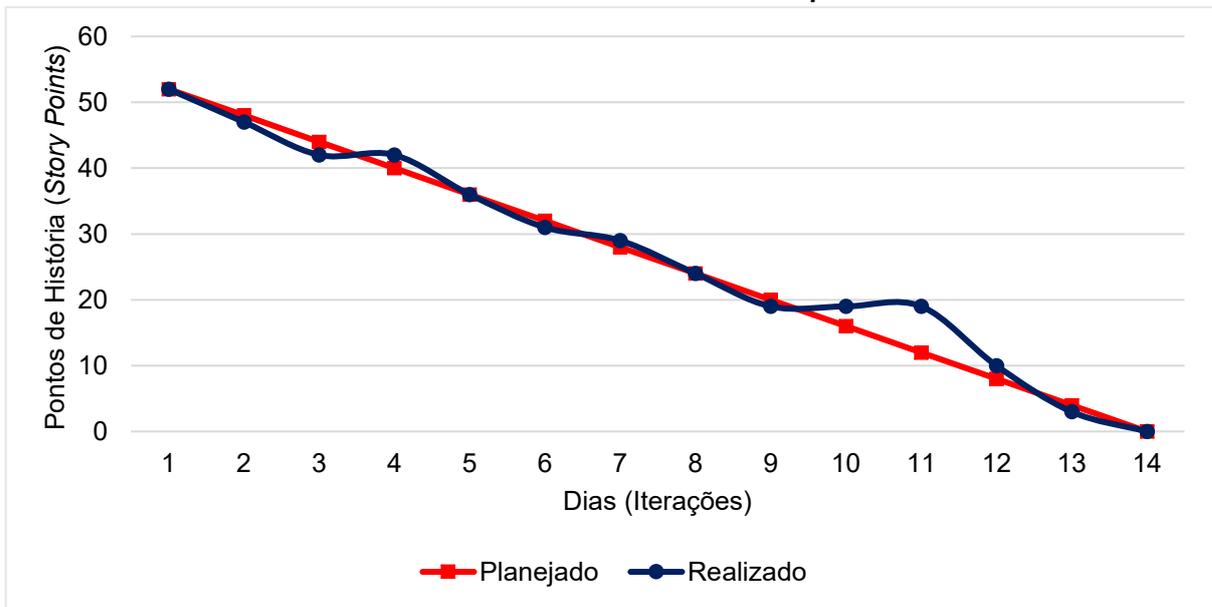
dessa etapa foi a mudança na paleta de cores do sistema. Especificam-se as atividades propostas e pontos de história atribuídos em função do nível de dificuldade enfrentado conforme o Quadro 44.

Quadro 44 – Atividades realizadas durante a *Sprint 5*

Atividades	Pontos
Corrigir os erros que foram identificados em atividades da <i>Sprint 4</i>	5
Trocar a paleta de cores do sistema	5
Criar a classe modelo para clientes	1
Criar uma tabela SQL para armazenar os clientes	2
Implementar o padrão DAO em Java para este modelo	2
Criar a tela principal de clientes	3
Criar a tela de cadastro de cliente	3
Criar um validador de CPF	5
Criar a tela de atualização de cliente	3
Validar o restante dos campos	2
Desenvolver os métodos de cadastro e atualização	2
Exibir os principais atributos em uma tabela de visualização do JavaFX	5
Exibir todos os atributos na seção de detalhes da tela principal	3
Criar o botão de excluir cliente	1
Programar a exibição de alertas para o usuário por meio de caixas de diálogo	2
Criar a barra de pesquisa de cliente e os botões de filtro	5
Realizar testes com o intuito de identificar problemas	3
Total	52

Fonte: Elaboração própria (2022).

O gráfico de *Burndown* gerado como consequência das baixas na pontuação pode ser visto no Gráfico 5. Analisando-o, entende-se que foi mais uma *Sprint* de sucesso, uma vez que todas as atividades planejadas foram concluídas a tempo. Seu andamento foi quase completamente estável, isto é, a linha de trabalho real permaneceu muito próxima à linha de trabalho ideal, com apenas uma pequena variação nos dias finais do desenvolvimento.

Gráfico 5 – Gráfico de *Burndown* da *Sprint* 5

Fonte: Elaboração própria (2022).

Sobre a retrospectiva da *Sprint* 5, realizada no dia 28 de maio de 2022, os seguintes aspectos foram considerados:

- a) O que deu certo: as atividades propostas foram pensadas sob medida visando as dificuldades que poderiam ser enfrentadas nos dias finais da *Sprint* (e que de fato foram), demonstrando assim uma melhora na capacidade de planejamento da equipe;
- b) O que deu errado: apesar de considerar uma importante parte do sistema – os clientes – não foi uma *Sprint* muito inovadora no quesito de funcionalidades, dado que apenas replica (com poucas diferenças) os métodos implementados no módulo de fornecedores (ambas as classes são semelhantes por natureza);
- c) Ações de melhorias: dedicar mais tempo ao projeto.

4.2.6 *Sprint* 6

A sexta etapa de desenvolvimento do projeto teve início em 30 de maio de 2022 e fim em 12 de junho. Uma vez que os módulos de produtos e de clientes foram finalizados, tornou-se possível iniciar o desenvolvimento das funcionalidades do sistema relacionadas às vendas da academia e itens contidos em cada venda.

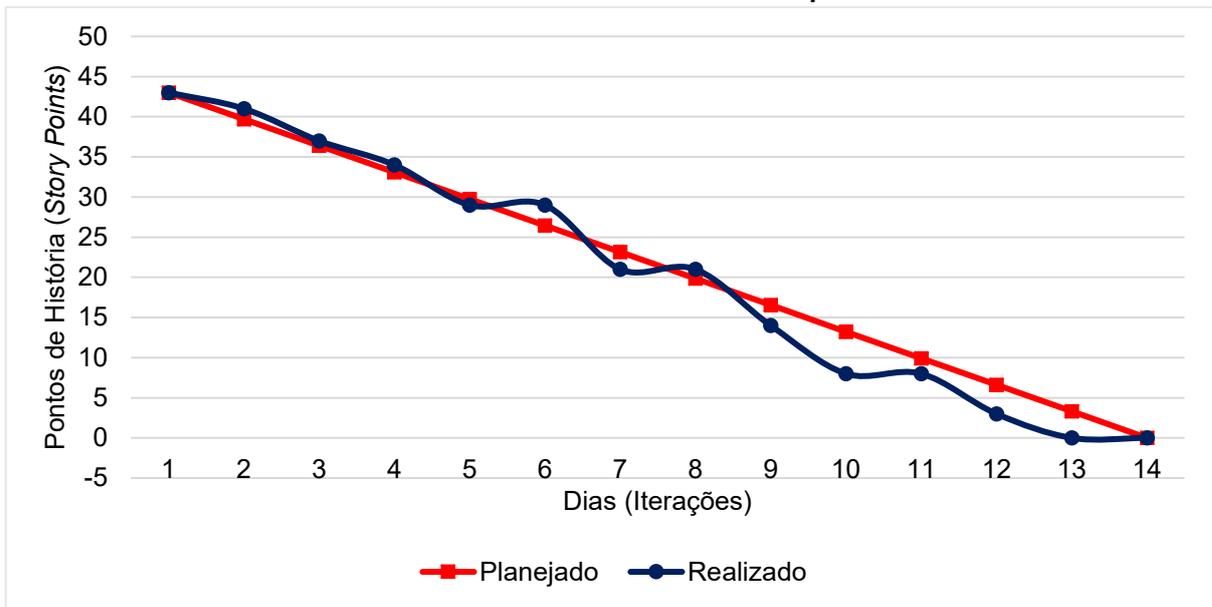
Especificam-se as atividades propostas e pontos de história atribuídos em função do nível de dificuldade enfrentado conforme o Quadro 45.

Quadro 45 – Atividades realizadas durante a *Sprint 6*

Atividades	Pontos
Criar a classe modelo para vendas	1
Criar a classe modelo para itens de venda	1
Criar uma tabela SQL para armazenar as vendas	2
Criar uma tabela SQL para armazenar os itens de venda	2
Criar <i>triggers</i> SQL para atualizar a quantidade do produto ao ser vendido	1
Implementar o padrão DAO em Java para estes modelos	2
Criar a tela principal de vendas	5
Criar a tela para inserir itens de venda	8
Validar os campos e cadastrar	2
Exibir os principais atributos em uma tabela de visualização do JavaFX	5
Exibir todos os atributos na seção de detalhes da tela principal	3
Criar o botão de cancelar venda	3
Programar a exibição de alertas para o usuário por meio de caixas de diálogo	2
Criar a barra de pesquisa de venda e os botões de filtro	3
Realizar testes com o intuito de identificar problemas	3
Total	43

Fonte: Elaboração própria (2022).

O gráfico de *Burndown* gerado como consequência das baixas na pontuação pode ser visto no Gráfico 6. Analisando-o, entende-se que foi outra *Sprint* de sucesso, visto que todas as atividades propostas foram finalizadas a tempo.

Gráfico 6 – Gráfico de *Burndown* da *Sprint* 6

Fonte: Elaboração própria (2022).

Sobre a retrospectiva da *Sprint*, realizada no dia anterior à data-limite, os seguintes tópicos foram considerados:

- a) O que deu certo: os integrantes conseguiram dedicar mais tempo ao desenvolvimento do sistema e à documentação;
- b) O que deu errado: foram enfrentadas algumas dificuldades para lidar com uma atualização do XAMPP, que precisou ser reinstalado e configurado novamente;
- c) Ações de melhorias: em relação ao XAMPP, manter uma cópia dos arquivos de configuração para que esse problema não aconteça novamente.

4.2.7 *Sprint* 7

A sétima etapa deste trabalho acadêmico teve início em 13 de junho de 2022 e terminou duas semanas depois, em 26 de junho. As atividades executadas durante esse período marcaram o começo do desenvolvimento do módulo de treinos do sistema da academia, o qual é dividido em exercícios, treinos e fichas de clientes. À vista disso, a *Sprint* 7 focou nas duas primeiras partes desse módulo: exercícios e treinos. O Quadro 46 apresenta, em detalhes, todas as atividades propostas e a

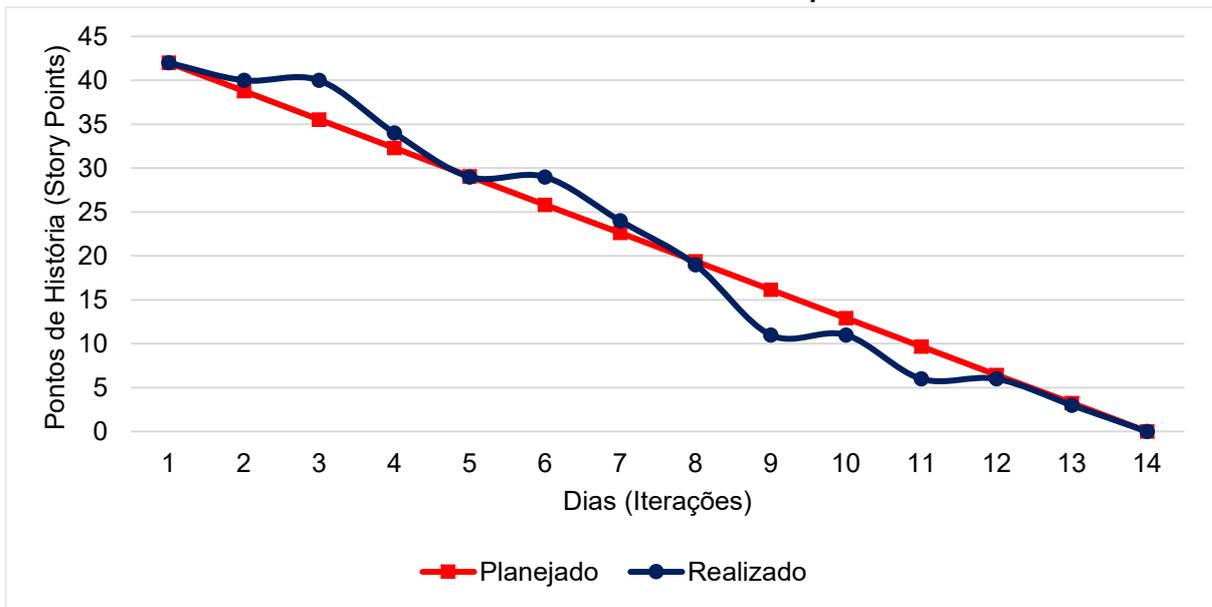
pontuação de cada uma em função do nível de dificuldade ou tempo requerido para a finalização.

Quadro 46 – Atividades realizadas durante a *Sprint 7*

Atividades	Pontos
Criar a classe modelo para exercícios	1
Criar a classe modelo para treinos	1
Criar uma tabela SQL para armazenar os exercícios	2
Criar uma tabela SQL para armazenar os treinos	2
Implementar o padrão DAO em Java para estes modelos	2
Criar as telas principais para cada modelo	5
Criar as telas de cadastro para cada modelo	5
Criar a tela de atualização de exercício	3
Validar os campos	2
Desenvolver os métodos de cadastro, atualização e exclusão	3
Exibir os principais atributos em tabelas de visualização do JavaFX	5
Exibir todos os atributos nas seções de detalhes de cada tela principal	3
Programar a exibição de alertas para o usuário por meio de caixas de diálogo	2
Criar a barra de pesquisa de exercício e de treino, e os botões de filtro	3
Realizar testes com o intuito de identificar problemas	3
Total	42

Fonte: Elaboração própria (2022).

A partir das baixas na pontuação, obteve-se outro gráfico de *Burndown*, presente logo abaixo, conforme o Gráfico 7. Analisando-o, entende-se que foi mais uma *Sprint* de sucesso, uma vez que todas as atividades planejadas foram concluídas a tempo.

Gráfico 7 – Gráfico de *Burndown* da *Sprint* 7

Fonte: Elaboração própria (2022).

Durante a reunião de retrospectiva da *Sprint*, que aconteceu um dia antes da data-limite, cada membro fez uma autoavaliação e ao final, juntos, consideraram as seguintes questões:

- a) O que deu certo: foi uma tentativa bem-sucedida de aumentar a carga de desenvolvimento, principalmente no que se refere à quantidade de telas desenvolvidas;
- b) O que deu errado: a princípio, na tela de cadastro de exercício, pensou-se na possibilidade de incluir um campo para o usuário carregar uma demonstração em vídeo de como realizar o exercício físico corretamente, porém, por questões técnicas, não foi possível ir adiante com a ideia;
- c) Ações de melhorias: aumentar a carga de desenvolvimento das próximas *Sprints*.

4.2.8 *Sprint* 8

Logo após o término da sétima *Sprint*, a equipe optou por aproveitar o tempo do recesso acadêmico para planejar melhor o restante das funcionalidades que ainda seriam implementadas no sistema. Assim sendo, o andamento do projeto teve uma pausa durante o mês de julho. A preparação das atividades para esta *Sprint* ocorreu

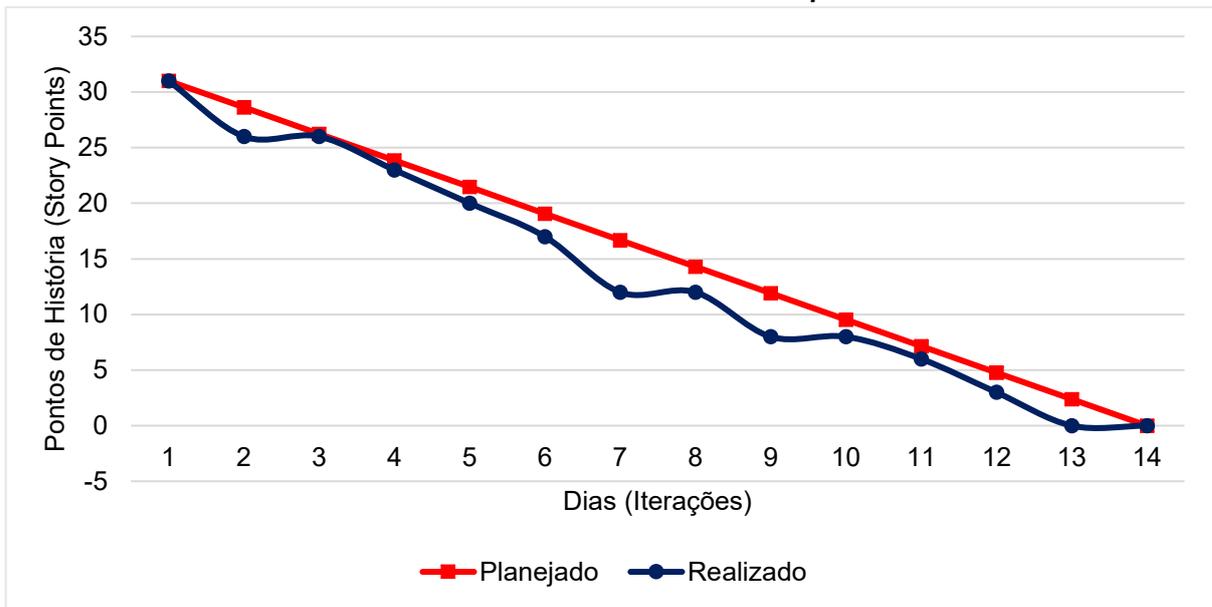
no dia 1 de agosto de 2022, com prazo de entrega para o dia 14 de agosto. Em suma, ficou estabelecido entre os participantes que a oitava etapa seria responsável por finalizar o módulo de treinos, sendo necessário, portanto, desenvolver as telas referentes às fichas de clientes da academia. O Quadro 47 apresenta, em detalhes, todas as atividades propostas bem como a pontuação de cada uma em função do nível de dificuldade ou tempo requerido para a finalização.

Quadro 47 – Atividades realizadas durante a *Sprint 8*

Atividades	Pontos
Criar a classe modelo para fichas de clientes	1
Criar uma tabela SQL para armazenar as fichas de clientes	2
Implementar o padrão DAO em Java para este modelo	2
Criar a tela principal de fichas	3
Criar a tela de cadastro de ficha	3
Validar os campos e cadastrar	3
Exibir os principais atributos em uma tabela de visualização do JavaFX	5
Exibir todos os atributos na seção de detalhes da tela principal	3
Criar o botão de excluir ficha	1
Programar a exibição de alertas para o usuário por meio de caixas de diálogo	2
Criar a barra de pesquisa de ficha e os botões de filtro	3
Realizar testes com o intuito de identificar problemas	3
Total	31

Fonte: Elaboração própria (2022).

A partir das baixas na pontuação, obteve-se outro gráfico de *Burndown*, que pode ser observado no Gráfico 8. Analisando-o, percebe-se que a linha de trabalho real se manteve abaixo da linha de trabalho ideal em quase todos os momentos desse período (ambas coincidem em alguns pontos, porém a primeira não chegou a superar a segunda).

Gráfico 8 – Gráfico de *Burndown* da *Sprint* 8

Fonte: Elaboração própria (2022).

Durante a reunião de retrospectiva da *Sprint*, que aconteceu um dia antes da data de entrega, cada membro fez uma autoavaliação e ao final, juntos, consideraram os seguintes aspectos:

- a) O que deu certo: foi a *Sprint* com a linha de trabalho real mais adiantada, evidenciando, dessa maneira, a importância da pausa no projeto para planejar melhor os próximos passos;
- b) O que deu errado: um problema envolvendo o phpMyAdmin teve de ser resolvido nesse período, já que o mesmo estava apresentando um erro de conflito com o navegador *web* (apesar disso, não prejudicou em nada o banco de dados do sistema);
- c) Ações de melhorias: criar um cronograma contendo o trabalho restante para guiar melhor cada integrante nos meses finais do projeto.

4.2.9 *Sprint* 9

A *Sprint* 9 de desenvolvimento do projeto teve início em 15 de agosto de 2022, com data de entrega para 28 de agosto. As atividades planejadas para esta etapa estão relacionadas com as funções de cadastro, leitura, atualização e exclusão das medidas dos perímetros corporais e das mensalidades dos clientes que frequentam a

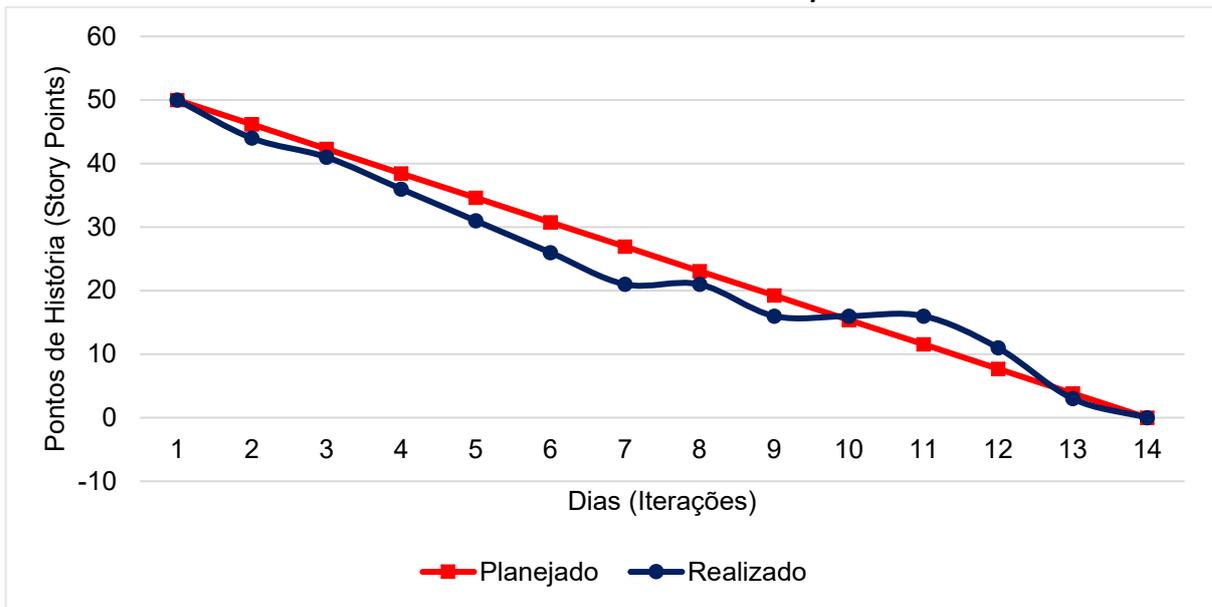
academia. O Quadro 48 expõe as atividades executadas e suas respectivas pontuações.

Quadro 48 – Atividades realizadas durante a *Sprint 9*

Atividades	Pontos
Criar a classe modelo para medidas	1
Criar a classe modelo para mensalidades	1
Criar uma tabela SQL para armazenar as medidas	2
Criar uma tabela SQL para armazenar as mensalidades	2
Implementar o padrão DAO em Java para estes modelos	3
Criar as telas principais para cada modelo	5
Criar as telas de cadastro para cada modelo	5
Criar as telas de atualização para cada modelo	5
Validar os campos	5
Desenvolver os métodos de cadastro, atualização e exclusão	5
Exibir os principais atributos em tabelas de visualização do JavaFX	5
Exibir todos os atributos nas seções de detalhes de cada tela principal	3
Programar a exibição de alertas para o usuário por meio de caixas de diálogo	2
Criar a barra de pesquisa de medida e de mensalidade, e os botões de filtro	3
Realizar testes com o intuito de identificar problemas	3
Total	50

Fonte: Elaboração própria (2022).

Durante as semanas de desenvolvimento, a equipe foi efetuando baixas nas atividades à medida em que elas eram concluídas, obtendo, ao final, mais um gráfico de Burndown, como pode ser observado no Gráfico 9.

Gráfico 9 – Gráfico de *Burndown* da *Sprint* 9

Fonte: Elaboração própria (2022).

Sobre a retrospectiva, o resumo de alguns tópicos discutidos encontra-se a seguir:

- a) O que deu certo: implementação de uma troca dinâmica de imagens com base no campo de texto selecionado para auxiliar o usuário nas telas de cadastro e de atualização de medidas (recurso que até então não havia sido utilizado);
- b) O que deu errado: em um determinado momento, o Scene Builder parou de carregar e exibir os componentes de uma biblioteca responsável pelos ícones do sistema. Para solucionar o problema, foi necessário reinstalar o *software*;
- c) Ações de melhorias: conscientização sobre o tempo restante para a conclusão do projeto.

4.2.10 *Sprint* 10

O planejamento da penúltima *Sprint* deste sistema de academia começou no dia 29 de agosto de 2022, com data-limite para o dia 11 de setembro. Para esta entrega, foram implementados os requisitos funcionais que diziam respeito ao módulo financeiro, ou seja, desenvolvimento da parte das cobranças (que são geradas a partir das vendas efetuadas e das mensalidades cadastradas) e dos pagamentos recebidos

referentes às próprias cobranças. O Quadro 49 apresenta as atividades realizadas e suas respectivas pontuações.

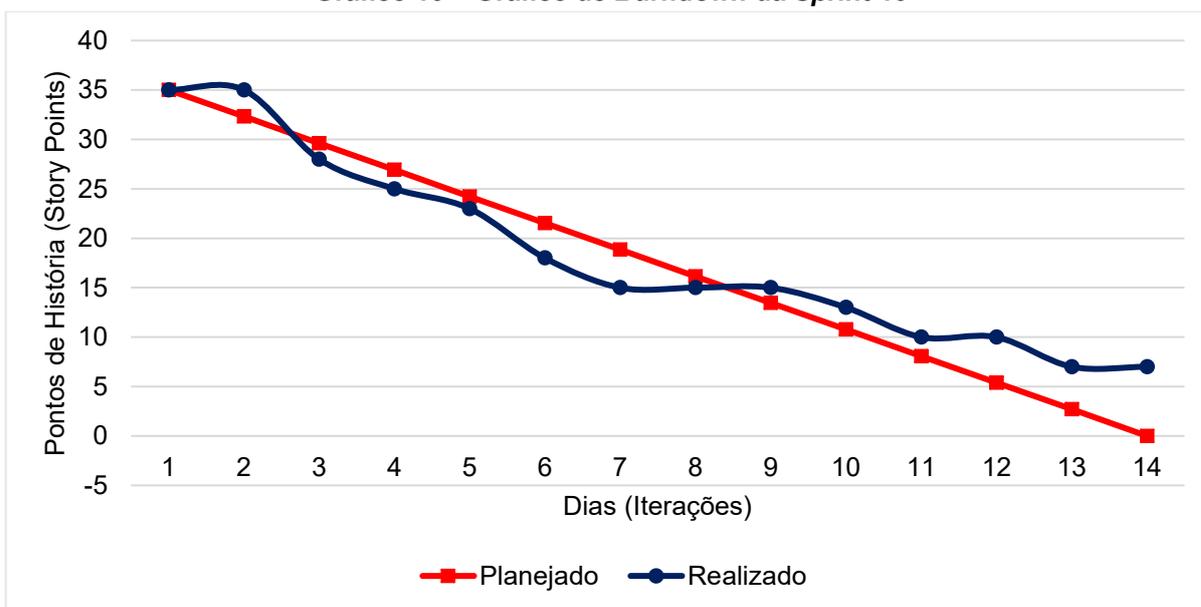
Quadro 49 – Atividades realizadas durante a *Sprint 10*

Atividades	Pontos
Criar a classe modelo para cobranças	1
Criar a classe modelo para pagamentos	1
Criar uma tabela SQL para armazenar as cobranças	2
Criar uma tabela SQL para armazenar os pagamentos	2
Implementar o padrão DAO em Java para estes modelos	3
Criar as telas principais para cada modelo	5
Criar a tela de recebimento de cobrança	3
Criar o botão para filtrar pelo tipo de cobrança	2
Exibir os principais atributos em tabelas de visualização do JavaFX	5
Exibir todos os atributos nas seções de detalhes de cada tela principal	3
Programar a exibição de alertas para o usuário por meio de caixas de diálogo	2
Criar a barra de pesquisa de cobrança e de pagamento, e os botões de filtro	3
Realizar testes com o intuito de identificar problemas	3
Total	35

Fonte: Elaboração própria (2022).

O gráfico de *Burndown* gerado como consequência das baixas na pontuação pode ser visto no Gráfico 10.

Gráfico 10 – Gráfico de *Burndown* da *Sprint 10*



Fonte: Elaboração própria (2022).

Nele, percebe-se que não foi possível concluir plenamente tudo que foi planejado em razão de algumas dificuldades com as classes do padrão DAO e com as tabelas SQL de cobranças e pagamentos, que estavam ocasionando muitas dúvidas procedimentais e gerando insatisfações, principalmente no que diz respeito às chaves estrangeiras. Desse modo, adiar a implementação dessas atividades para a próxima *Sprint* foi a melhor opção.

Sobre a retrospectiva dessa etapa, os seguintes tópicos foram considerados:

- a) O que deu certo: aumento da dedicação para a finalização do projeto;
- b) O que deu errado: as classes do padrão DAO e as tabelas SQL referentes às cobranças e aos pagamentos não foram implementadas da melhor maneira possível nesta *Sprint*;
- c) Ações de melhorias: refazer os diagramas de cobranças e pagamentos.

4.2.11 *Sprint* 11

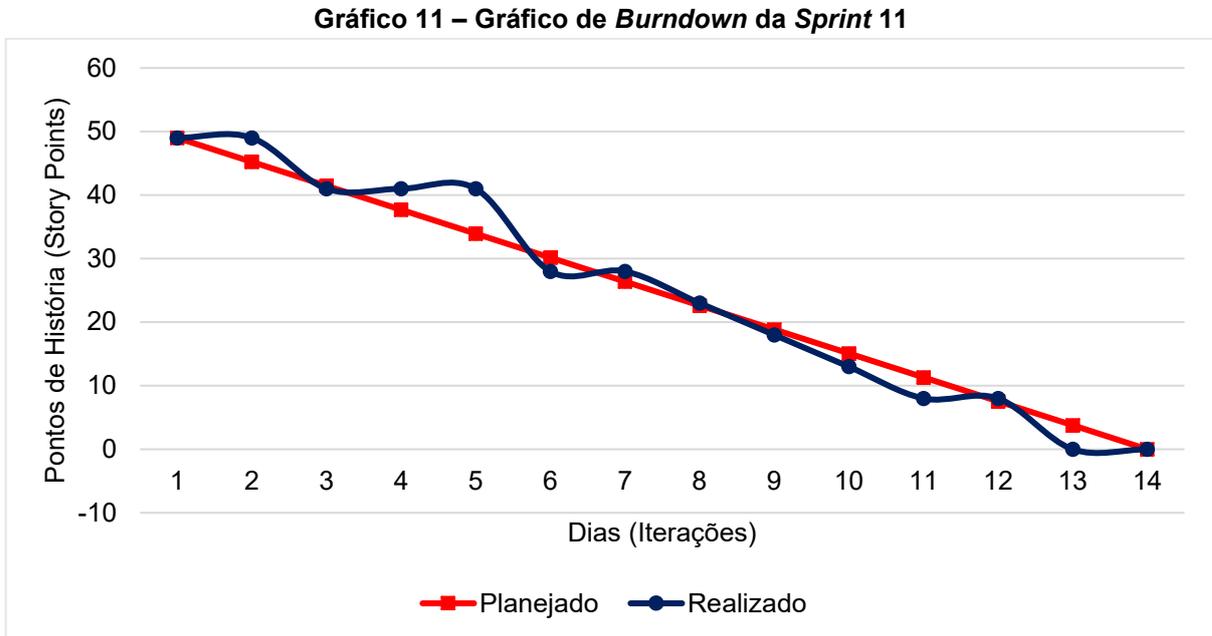
A última etapa de desenvolvimento começou no dia 12 de setembro de 2022 e terminou no dia 25 de setembro. Como descrito anteriormente, algumas atividades ficaram pendentes e precisaram ser incluídas nesta *Sprint*. Uma vez finalizando a implementação dessas atividades, todos os requisitos funcionais descritos no início do projeto estariam completos, tornando possível, dessa maneira, focar os esforços em outros aspectos para agregar na experiência do usuário, como exportar os dados para planilhas do Microsoft Excel e criar *dashboards*. As atividades planejadas bem como os pontos de história são apresentados a seguir, no Quadro 50.

Quadro 50 – Atividades realizadas durante a *Sprint* 11

Atividades	Pontos
Implementar as atividades pendentes da <i>Sprint</i> 10	8
Exportar os dados das tabelas de visualização para o Microsoft Excel	13
Criar uma <i>dashboard</i> com os dados dos clientes	5
Criar uma <i>dashboard</i> com os dados dos fornecedores	5
Criar uma <i>dashboard</i> com os dados dos produtos	5
Criar uma <i>dashboard</i> com os dados das vendas	5
Revisar mais uma vez todas as funcionalidades desenvolvidas	8
Total	49

Fonte: Elaboração própria (2022).

O último gráfico de *Burndown* gerado encontra-se abaixo, no Gráfico 11, no qual é possível observar que tudo ocorreu bem durante essas duas semanas.



Fonte: Elaboração própria (2022).

Na reunião de retrospectiva, os integrantes debateram, pela última vez, os acertos e os erros da *Sprint*. Justamente por ser a última etapa de desenvolvimento, o tópico de melhorias não foi considerado. O resumo do debate encontra-se a seguir:

- a) O que deu certo: as pendências da *Sprint* anterior foram resolvidas; muitas consultas SQL bem como trechos de código-fonte em Java complexos foram criados com sucesso, demonstrando assim uma grande e significativa evolução para o raciocínio lógico e outras capacidades cognitivas;
- b) O que deu errado: no início, a intenção era desenvolver as *dashboards* utilizando uma biblioteca criada por terceiros específica para isso, a qual fazia uso de blocos de painéis prontos para facilitar esse processo, contudo não foi possível implementá-la.

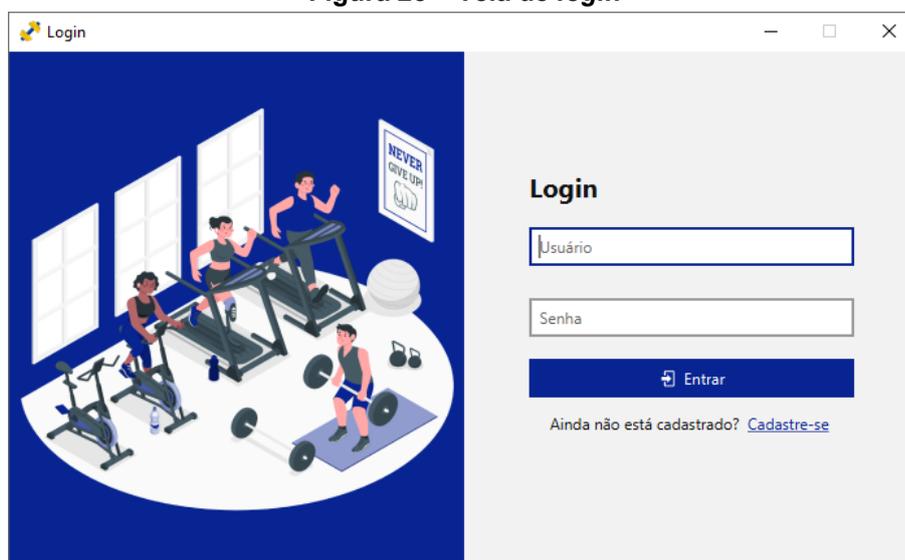
4.3 Interfaces

Uma interface gráfica de usuário é o meio pelo qual o usuário final interage ou controla um produto, *software* ou dispositivo de *hardware*. Sua função é tornar a interação humano-computador o mais suave, fácil e agradável possível.

Por uma questão de escolha, as interfaces deste sistema e seus componentes visuais fizeram uso de uma biblioteca do JavaFX criada por Vieira (2018) chamada JMetro, que contém diretrizes de estilo inspiradas no *Fluent*, um sistema de design *open-source* da Microsoft. Desse modo, o sistema possui um tema claro, com tons de cinza para as cores do plano de fundo e dos controles comuns, e uma cor de destaque para transmitir informações de estado, que nesse caso é o azul escuro encontrado no logo da academia BoxGym. Todas as telas que serão mostradas a seguir foram obtidas a partir do sistema em funcionamento utilizando uma ferramenta de captura.

A primeira tela com a qual o usuário se depara ao executar o sistema é a tela de *login*, conforme a Figura 25. Para entrar no sistema, é preciso informar o nome de usuário e senha previamente cadastrados e pressionar o botão “Entrar”. Se o usuário não possuir um cadastro, pode-se realizar um ao clicar no *link* “Cadastre-se”.

Figura 25 – Tela de *login*

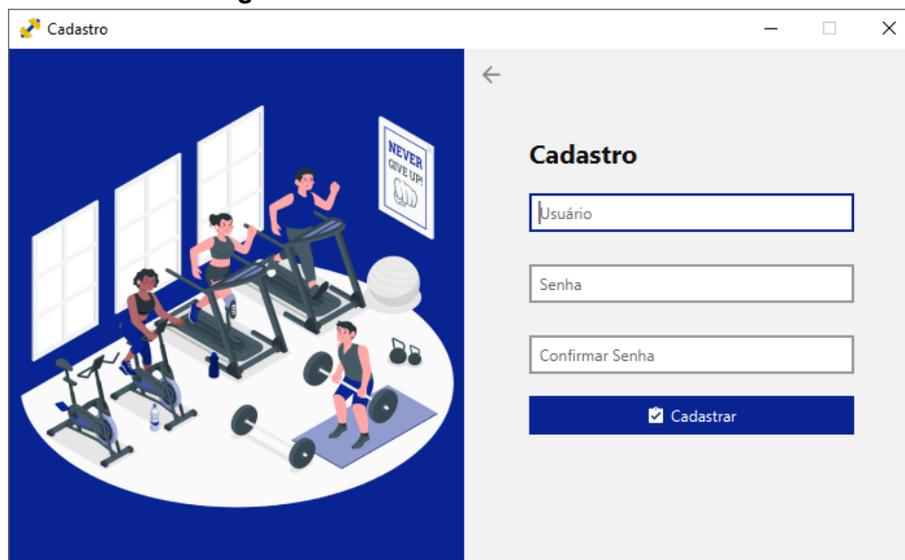


Fonte: Elaboração própria (2022).

Ao clicar no *link*, o indivíduo será redirecionado à tela de cadastro, apresentada na Figura 26. Nela, o usuário deve informar um nome de usuário e uma senha, confirmando-a em seguida. Para finalizar o cadastro e retornar automaticamente ao

login, basta clicar no botão “Cadastrar”. Caso opte por não prosseguir com o cadastro, o usuário tem a opção de voltar ao *login* clicando no ícone de seta localizado no canto superior esquerdo da segunda metade da tela.

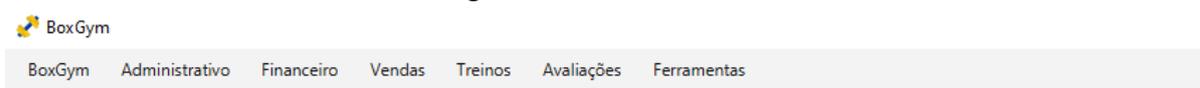
Figura 26 – Tela de cadastro de usuário



Fonte: Elaboração própria (2022).

Uma vez tendo realizado o *login*, o usuário se depara com a barra de menu da Figura 27 presente no topo do sistema.

Figura 27 – Barra de menu



Fonte: Elaboração própria (2022).

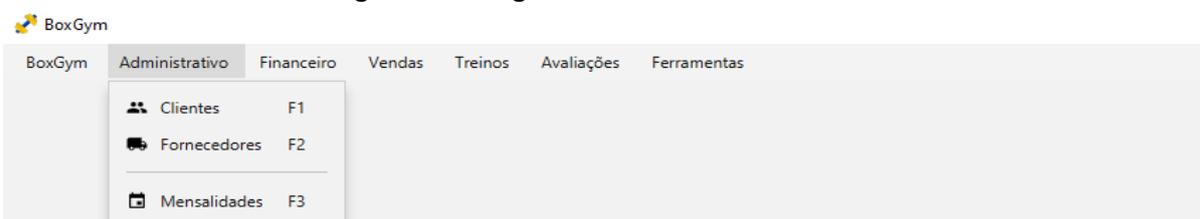
Os itens da barra de menu são compostos por um ou mais subitens, os quais possibilitam acessar as diferentes telas do sistema e suas funcionalidades. Cada subitem pode ser acessado através do clique do *mouse*, ou quando possível, utilizando atalhos, como as teclas HOME, ESC e as de função presentes na parte superior do teclado (de F1 até F12).

No primeiro item do menu, denominado “BoxGym” e ilustrado na Figura 28, há dois subitens: o “Início” possibilita voltar à tela inicial do sistema, que contém as *dashboards*; o “Sair”, como o próprio nome sugere, permite sair do sistema e voltar ao *login*.

Figura 28 – Primeiro item da barra de menu

Fonte: Elaboração própria (2022).

No segundo item do menu, denominado “Administrativo” e ilustrado na Figura 29, há três subitens: “Clientes”, “Fornecedores” e “Mensalidades”.

Figura 29 – Segundo item da barra de menu

Fonte: Elaboração própria (2022).

No terceiro item do menu, denominado “Financeiro” e ilustrado na Figura 30, há dois subitens: “Cobranças” e “Pagamentos”.

Figura 30 – Terceiro item da barra de menu

Fonte: Elaboração própria (2022).

No quarto item do menu, denominado “Vendas” e ilustrado na Figura 31, há três subitens: “Vendas”, “Produtos” e “Entradas de Estoque”.

Figura 31 – Quarto item da barra de menu

Fonte: Elaboração própria (2022).

No quinto item do menu, denominado “Treinos” e ilustrado na Figura 32, há três subitens: “Fichas”, “Treinos” e “Exercícios”.

Figura 32 – Quinto item da barra de menu



Fonte: Elaboração própria (2022).

No sexto item do menu, denominado “Avaliações” e ilustrado na Figura 33, há apenas o subitem “Medidas”.

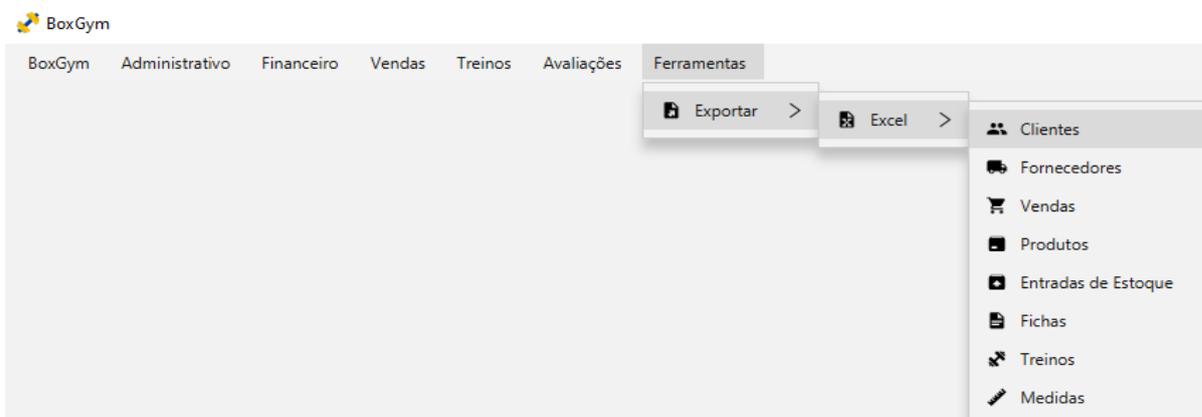
Figura 33 – Sexto item da barra de menu



Fonte: Elaboração própria (2022).

No último item do menu, denominado “Ferramentas” e ilustrado na Figura 34, há apenas o subitem “Exportar”, que permite a exportação dos registros cadastrados pelo usuário (e que estão armazenados nas tabelas do banco de dados) para planilhas do Microsoft Excel.

Figura 34 – Sétimo item da barra de menu



Fonte: Elaboração própria (2022).

Após a apresentação dos itens da barra de menu, torna-se fundamental apresentar as telas de cada subitem. Nesse sentido, começando pelos subitens do “Administrativo”, a Figura 35 exibe a tela de gerenciamento de clientes.

O primeiro elemento a ser destacado dessa tela é a tabela de visualização, um controle disponível no JavaFX projetado para apresentar um número ilimitado de linhas de dados, divididas em colunas. A tabela é preenchida com os registros dos clientes que retornam do banco de dados. Se não houver nenhum cliente cadastrado, a tabela permanece vazia. O segundo elemento, localizado à direita da tabela, é um painel de detalhes que contém todos os atributos da entidade *Customer*, os quais são preenchidos com os dados da linha selecionada da tabela. Por último, em cima da tabela, é possível observar cinco controles com os quais o usuário pode interagir. Da esquerda para a direita, os dois primeiros (o botão com opções de filtro e o campo de texto) estão relacionados com o mecanismo de pesquisa. Já os outros três botões são responsáveis pelas funções de cadastrar, atualizar e excluir. Uma observação importante: os registros presentes nessa e nas telas que ainda serão apresentadas são fictícios, isto é, foram utilizados apenas para fins de teste e demonstração.

Figura 35 – Tela de gerenciamento de clientes

CPF	Nome	Sexo	Data de Nascimento	E-mail	Tel
56401123010	Julio Victor Gomes	Masculino	12/02/1988	juliov.gomes51@outlook.com	1
71560109050	Felipe Anthony Novaes	Masculino	11/04/1990	felipeanthony874@gmail.com	1
56665597024	Livia Nicole Antonella Galvão	Feminino	10/05/2000	livia_antonella41@gmail.com	1
49224380093	Maya Bruna Milena da Cunha	Feminino	04/11/2000	mayabruna741@gmail.com	1
67572344089	Luiza Antônia de Paula	Feminino	07/11/2000	luizadepaula@gmail.com	1
08099734050	Sabrina Barbosa da Silva	Feminino	21/11/1995	sab_barbosa@gmail.com	1
99387884082	Lucas Oliveira Moura	Masculino	22/12/1996	lucasmoura@gmail.com	1
69212797075	Marcos Azevedo Vieira	Masculino	23/01/1996	azevedomarcos@outlook.com	1
27305043036	Danilo Fernandes Conceição	Masculino	04/05/1996	fernandes_danilo01@outlook.com	1
22472886004	Henrique Cavalline Pires	Masculino	06/05/1997	henrique_pires@outlook.com	1
83592215027	Luiz Miguel Viana	Masculino	11/04/1998	luizmiguelviana@outlook.com	1
23928493078	Pedro Lucas Cavalcanti	Masculino	19/04/1999	pedroLucascavalc@outlook.com	1
83405241006	Rafael das Neves	Masculino	20/01/2003	rafaelneves65@outlook.com	1
39567906009	Maria Vitória Ferreira	Feminino	01/07/2003	mariavitoria@gmail.com	1
82355582033	Sophia Novaes Rocha	Feminino	04/09/2002	sophynvroch10@outlook.com	1
96945962547	Fernando Oliver Souza	Masculino	02/01/1991	fernando1991@outlook.com	1

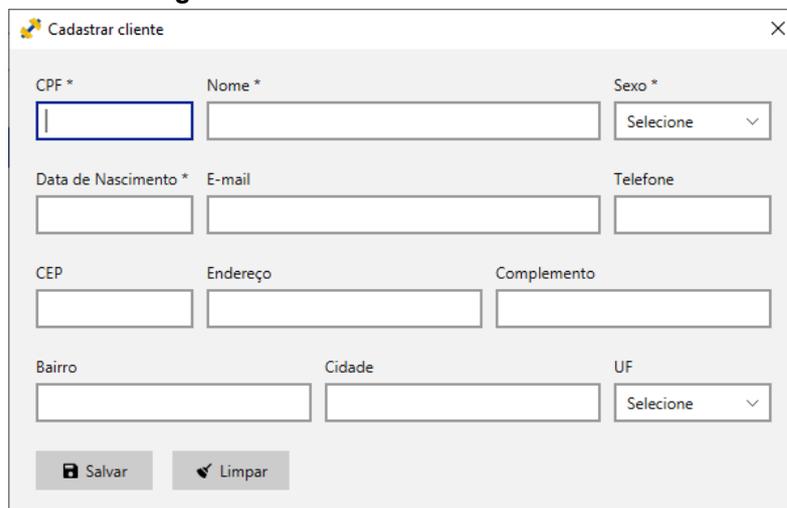
Detalhes	
ID:	1
CPF:	56401123010
Nome:	Julio Victor Gomes
Sexo:	Masculino
Data de Nascimento:	12/02/1988
E-mail:	juliov.gomes51@outlook.com
Telefone:	19996918907
CEP:	13385004
Endereço:	R. Anchieta, 875
Complemento:	
Bairro:	Jardim Bela Vista
Cidade:	Nova Odessa
UF:	SP
Criação:	01/08/2022 18:20:02
Modificação:	01/08/2022 18:20:02

Fonte: Elaboração própria (2022).

Ao clicar no botão “Cadastrar”, a janela da Figura 36 é aberta. Para realizar um cadastro, o usuário do sistema deve obrigatoriamente informar o CPF, nome, sexo e

data de nascimento do cliente a ser cadastrado (os demais campos são opcionais), e clicar em “Salvar”. Feito isso, a janela se fecha, o cliente é inserido no banco de dados e logo em seguida aparece na tabela de visualização da tela de gerenciamento.

Figura 36 – Tela de cadastro de clientes

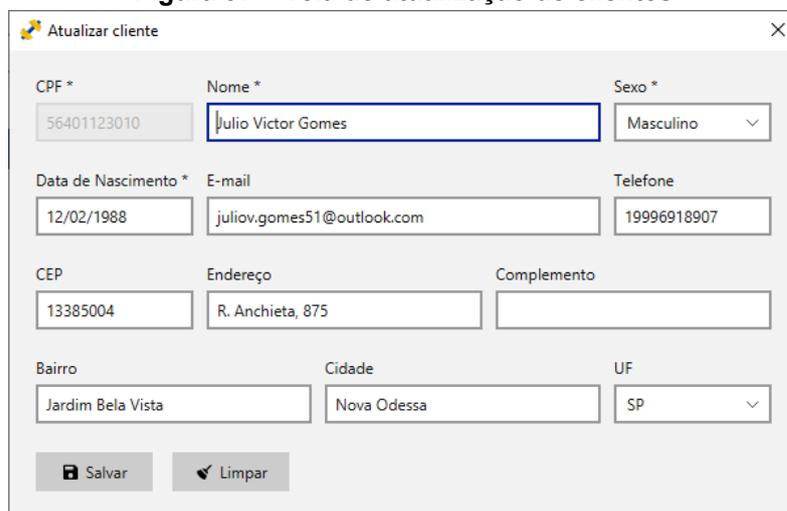


The screenshot shows a web form titled "Cadastrar cliente" with a close button (X) in the top right corner. The form contains several input fields and dropdown menus arranged in a grid. The fields are: CPF * (empty), Nome * (empty), Sexo * (dropdown menu with "Selecione" selected), Data de Nascimento * (empty), E-mail (empty), Telefone (empty), CEP (empty), Endereço (empty), Complemento (empty), Bairro (empty), Cidade (empty), and UF (dropdown menu with "Selecione" selected). At the bottom of the form, there are two buttons: "Salvar" (Save) and "Limpar" (Clear).

Fonte: Elaboração própria (2022).

Caso seja necessário atualizar o cadastro de algum cliente, o usuário do sistema deve selecioná-lo na tabela de visualização e clicar no botão “Atualizar”. Ao fazer isso, uma janela como a da Figura 37 é aberta com todos os dados previamente cadastrados daquele cliente para que possam ser modificados. Para finalizar, basta clicar no botão “Salvar”.

Figura 37 – Tela de atualização de clientes



The screenshot shows a web form titled "Atualizar cliente" with a close button (X) in the top right corner. The form contains several input fields and dropdown menus, all of which are pre-filled with data. The fields are: CPF * (56401123010), Nome * (Julio Victor Gomes), Sexo * (dropdown menu with "Masculino" selected), Data de Nascimento * (12/02/1988), E-mail (juliov.gomes51@outlook.com), Telefone (19996918907), CEP (13385004), Endereço (R. Anchieta, 875), Complemento (empty), Bairro (Jardim Bela Vista), Cidade (Nova Odessa), and UF (dropdown menu with "SP" selected). At the bottom of the form, there are two buttons: "Salvar" (Save) and "Limpar" (Clear).

Fonte: Elaboração própria (2022).

A respeito do segundo subitem do “Administrativo”, a Figura 38 apresenta a tela de gerenciamento de fornecedores. Do mesmo modo que a tela de gerenciamento anterior, ela é composta por uma tabela de visualização, carregada com os registros dos fornecedores que retornam do banco de dados; um painel de detalhes, que contém os atributos da entidade *Supplier* em sua totalidade, os quais são preenchidos com os dados da linha selecionada da tabela; e cinco controles de interação.

Figura 38 – Tela de gerenciamento de fornecedores

CNPJ	Razão Social	Nome Fantasia	E-mail	Telefone
09339936000116	Ns2.com Internet S.A.	Netshoes	fiscal.estadual@magazineleza.com.br	1637112
06347409000165	SBF Comércio de Produtos Esportivos ...	Centauro	grupofiscal@centauro.com.br	1123840
10832644000108	Growth Supplements - Produtos Alime...	Growth Supplements	fiscal@growthsupplements.com.br	4733690
13807794000141	Espartanos Comércio de Artigos Espor...	Espartanos	administrativo@gmaf.com.br	1198911
35818800000107	Led Comércio de Suplementos Eireli	Madrugão Suplementos	sac@madrugaosuplementos.com.br	1140635
45211717000110	Suplemento Curitiba Afjr LTDA	Suplementos Curitiba	info@suplementoscuritiba.com.br	4130958
57235426000303	Brg Suplementos Nutricionais LTDA	Integralmédica	maria.santos@integralmedica.com.br	1146627
21519597000155	Nutrifit Comercial Pinheiros LTDA	Nutrifit	sac@lojanutrifit.com.br	1130312
18784087000109	BT1 Suplementos Alimentares LTDA	Betta Suplementos	atendimento@bettasuplementos.com.br	1941223
22347552000103	Associação Sociedade da Caveira Preta	Black Skull	primordialc@gmail.com	6132012
07578713000429	Supley Lab. de Alimentos e Suplement...	Max Titanium	maxtitanium@maxtitanium.com.br	1635062
08998124000110	3VS Nutrition	3VS Nutrition	sac@3vsnutrition.com.br	1194305
17218436000154	Metamorfose Brasil	Metamorfose Brasil	contato@metamorfosebrasil.com	2499988
14871851000114	Body Builders Suplementação Esportiv...	Body Builders	sac@bodybuilders.com.br	1195366
42274696002561	adidas do Brasil LTDA	Adidas	atendimentoadidas@adidas.com.br	
17788352000156	Bulkino Indústria e Comércio de Roup...	Bulkino	contato@bulkino.com.br	6231105

Exibindo 16 resultados Linha 1 selecionada

Detalhes

ID: 1

CNPJ: 09339936000116

Razão Social: Ns2.com Internet S.A.

Nome Fantasia: Netshoes

E-mail: fiscal.estadual@magazineleza.com.br

Telefone: 1637112002

CEP: 02047901

Endereço: R. Maria Prestes Maia, 300

Complemento: Andar 2

Bairro: Carandiru

Cidade: São Paulo

UF: SP

Criação: 01/08/2022 19:20:02

Modificação: 01/08/2022 19:20:02

Fonte: Elaboração própria (2022).

Ao clicar no botão “Cadastrar”, a janela da Figura 39 pode ser vista. Para realizar um cadastro, o usuário do sistema deve obrigatoriamente informar o CNPJ, a razão social e o nome fantasia do fornecedor a ser cadastrado; os demais campos são opcionais.

Figura 39 – Tela de cadastro de fornecedores

Cadastrar fornecedor

CNPJ * Razão Social *

Nome Fantasia *

E-mail Telefone

CEP Endereço Complemento

Bairro Cidade UF
Selecione

Salvar Limpar

Fonte: Elaboração própria (2022).

Caso seja necessário atualizar o cadastro de algum fornecedor, o usuário do sistema deve selecioná-lo na tabela de visualização e clicar no botão “Atualizar”. Ao fazer isso, uma janela como a da Figura 40 é aberta com todos os dados previamente cadastrados daquele fornecedor para que possam ser modificados.

Figura 40 – Tela de atualização de fornecedores

Atualizar fornecedor

CNPJ * Razão Social *

Nome Fantasia *

E-mail Telefone

CEP Endereço Complemento

Bairro Cidade UF
SP

Salvar Limpar

Fonte: Elaboração própria (2022).

O último subitem desta primeira parte do menu é a tela de gerenciamento de mensalidades, conforme a Figura 41. Em sua estrutura, é possível observar os seguintes componentes: tabela de visualização, painel de detalhes que contém os atributos da entidade *Membership*, mecanismo de pesquisa, caixa de seleção para mudar a visualização dos dados na tabela (exibir todas as mensalidades, somente as pagas ou somente as inadimplentes) e três botões (cadastrar, atualizar e cancelar).

Figura 41 – Tela de gerenciamento de mensalidades

Cliente	Data de Vencimento	Valor	Status
Julio Victor Gomes	01/10/2022	R\$ 80,00	Pago
Julio Victor Gomes	01/11/2022	R\$ 80,00	Pago
Julio Victor Gomes	01/12/2022	R\$ 80,00	Aguardando pagamento
Julio Victor Gomes	01/01/2023	R\$ 80,00	Aguardando pagamento
Julio Victor Gomes	01/02/2023	R\$ 80,00	Aguardando pagamento
Julio Victor Gomes	01/03/2023	R\$ 80,00	Aguardando pagamento
Sabrina Barbosa da Silva	05/10/2022	R\$ 80,00	Pago
Sabrina Barbosa da Silva	05/11/2022	R\$ 80,00	Pago
Sabrina Barbosa da Silva	05/12/2022	R\$ 80,00	Aguardando pagamento
Lucas Oliveira Moura	25/11/2022	R\$ 80,00	Aguardando pagamento
Lucas Oliveira Moura	25/12/2022	R\$ 80,00	Aguardando pagamento
Lucas Oliveira Moura	25/01/2023	R\$ 80,00	Aguardando pagamento

Exibindo 12 resultados Linha 1 selecionada

Detalhes

ID: 1
 Cliente: Julio Victor Gomes
 Data de Vencimento: 01/10/2022
 Valor: R\$ 80,00
 Status: Pago
 Criação: 01/10/2022 18:41:33
 Modificação: 01/10/2022 18:41:33

Fonte: Elaboração própria (2022).

O cadastro de uma ou mais mensalidades para um determinado cliente pode ser feito por meio da interface da Figura 42. Para tal, é preciso obrigatoriamente informar o nome do cliente, a data de vencimento, o valor a ser pago e a quantidade de mensalidades que serão geradas (correspondem, em meses, ao tempo que o cliente irá frequentar a academia). Ao preencher os campos e pressionar o botão “Salvar”, além de inserir a(s) mensalidades(s) na tabela SQL, o sistema também irá gerar a(s) respectiva cobrança(s), inserindo-a(s), igualmente, no banco de dados.

Figura 42 – Tela de cadastro de mensalidades

A captura de tela mostra uma janela intitulada "Cadastrar mensalidade(s)". No topo, há um ícone de seta amarela apontando para cima e um botão de fechar "X". Abaixo, há quatro campos de entrada: "Cliente *" com um menu suspenso contendo "Selecione"; "Data de Vencimento *" com o texto "Escolha uma data" e um ícone de calendário; "Valor *" com o valor "R\$ 0,00"; e "Quantidade *" com um menu suspenso contendo "Selecione". Na base da janela, há dois botões: "Salvar" com um ícone de disquete e "Limpar" com um ícone de lixeira.

Fonte: Elaboração própria (2022).

Para atualizar os dados de alguma mensalidade, o usuário deve selecioná-la na tabela de visualização e clicar em “Atualizar”. Após essa ação, uma tela como a da Figura 43 irá aparecer contendo os dados previamente cadastrados.

Figura 43 – Tela de atualização de mensalidades

A captura de tela mostra uma janela intitulada "Atualizar mensalidade". No topo, há um ícone de seta amarela apontando para cima e um botão de fechar "X". Abaixo, há quatro campos de entrada: "Cliente *" com um menu suspenso contendo "Julio Victor Gomes"; "Data de Vencimento *" com o valor "01/12/2022" e um ícone de calendário; "Valor *" com o valor "R\$ 80,00"; e "Quantidade *" com um menu suspenso vazio. Na base da janela, há dois botões: "Salvar" com um ícone de disquete e "Limpar" com um ícone de lixeira.

Fonte: Elaboração própria (2022).

O próximo subitem, localizado no item de menu “Financeiro”, é responsável por conter os valores a receber dos clientes da academia, gerados com base nas mensalidades cadastradas e vendas efetuadas. As cobranças abertas podem ser visualizadas na tela de gerenciamento de duas formas conforme a opção da caixa de seleção: visualizar apenas as cobranças referentes às mensalidades (Figura 44) ou apenas as cobranças referentes às vendas (Figura 45).

Figura 44 – Tela de gerenciamento das mensalidades a receber

Financeiro > Cobranças

Exibindo 8 resultados Linha 1 selecionada

Cliente	Descrição	Data de Vencimento	Valor
Julio Victor Gomes	Mensalidade 3/6	01/12/2022	R\$ 80,00
Julio Victor Gomes	Mensalidade 4/6	01/01/2023	R\$ 80,00
Julio Victor Gomes	Mensalidade 5/6	01/02/2023	R\$ 80,00
Julio Victor Gomes	Mensalidade 6/6	01/03/2023	R\$ 80,00
Sabrina Barbosa da Silva	Mensalidade 3/3	05/12/2022	R\$ 80,00
Lucas Oliveira Moura	Mensalidade 1/3	25/11/2022	R\$ 80,00
Lucas Oliveira Moura	Mensalidade 2/3	25/12/2022	R\$ 80,00
Lucas Oliveira Moura	Mensalidade 3/3	25/01/2023	R\$ 80,00

Detalhes

ID: 20
Cliente: Julio Victor Gomes
Descrição: Mensalidade 3/6
Data de Vencimento: 01/12/2022
Valor: R\$ 80,00
Criação: 07/11/2022 18:40:12
Modificação: 07/11/2022 18:40:12

Receber Cobrança

Fonte: Elaboração própria (2022).

Figura 45 – Tela de gerenciamento das vendas a receber

Financeiro > Cobranças

Exibindo 3 resultados Linha 1 selecionada

Cliente	Descrição	Data de Vencimento	Valor
Marcos Azevedo Vieira	Venda de Produtos	16/11/2022	R\$ 131,00
Livia Nicole Antonella Galvão	Venda de Produtos	21/11/2022	R\$ 59,99
Luiza Antônia de Paula	Venda de Produtos	23/11/2022	R\$ 164,97

Detalhes

ID: 15
Cliente: Marcos Azevedo Vieira
Descrição: Venda de Produtos
Data de Vencimento: 16/11/2022
Valor: R\$ 131,00
Criação: 16/11/2022 20:53:33
Modificação: 16/11/2022 20:53:33

Receber Cobrança

Fonte: Elaboração própria (2022).

Ao selecionar uma cobrança na tabela, é possível registrar o seu pagamento clicando no botão “Receber Cobrança”, localizado no painel de detalhes. Ao fazer isso, uma janela como a da Figura 46 aparece, em que o usuário deve informar a data de recebimento e o valor a ser pago pelo cliente, podendo ser o valor total em dinheiro

ou apenas parte dele. Em ambos os casos, o valor recebido é registrado na tabela de pagamentos. No primeiro, a cobrança é finalizada; no segundo, entretanto, a cobrança continua aberta, mas com o devido ajuste em seu valor total (obtido a partir da subtração entre o valor a pagar e o parcial recebido).

Figura 46 – Tela de recebimento de mensalidades e vendas

Receber pagamento

Data de Recebimento * 23/11/2022

Dinheiro * 131,00

Valor a Pagar 131,00

Recebendo 131,00

0,00 =

Salvar

Fonte: Elaboração própria (2022).

O último subitem da parte financeira do sistema, que pode ser contemplado na Figura 47, contém todos os pagamentos recebidos de mensalidades e vendas (totais e parciais).

Figura 47 – Tela de visualização de pagamentos

BoxGym

Administrativo Financeiro Vendas Treinos Avaliações Ferramentas

Financeiro > Pagamentos

Pesquisar...

Cliente	Descrição	Data de Pagamento	Valor da Cobrança	Valor Pago
Julio Victor Gomes	Venda de Produtos	16/08/2022	R\$ 131,00	R\$ 131,00
Lucas Oliveira Moura	Venda de Produtos	18/08/2022	R\$ 45,00	R\$ 45,00
Sabrina Barbosa da Silva	Venda de Produtos	30/08/2022	R\$ 45,00	R\$ 45,00
Danilo Fernandes Conceição	Venda de Produtos	01/09/2022	R\$ 29,99	R\$ 29,99
Marcos Azevedo Vieira	Venda de Produtos	07/09/2022	R\$ 35,00	R\$ 35,00
Danilo Fernandes Conceição	Venda de Produtos	20/09/2022	R\$ 135,00	R\$ 135,00
Livia Nicole Antonella Galvão	Venda de Produtos	05/10/2022	R\$ 270,00	R\$ 270,00
Pedro Lucas Cavalcanti	Venda de Produtos	10/10/2022	R\$ 119,98	R\$ 119,98
Lucas Oliveira Moura	Venda de Produtos	26/10/2022	R\$ 196,50	R\$ 196,50
Julio Victor Gomes	Parte de Vendas	01/11/2022	R\$ 249,60	R\$ 200,00
Julio Victor Gomes	Parte de Vendas	02/11/2022	R\$ 49,60	R\$ 40,00
Julio Victor Gomes	Venda de Produtos	03/11/2022	R\$ 9,60	R\$ 9,60
Julio Victor Gomes	Mensalidade 1/6	07/11/2022	R\$ 80,00	R\$ 80,00
Julio Victor Gomes	Mensalidade 2/6	07/11/2022	R\$ 80,00	R\$ 80,00
Sabrina Barbosa da Silva	Mensalidade 1/3	07/11/2022	R\$ 80,00	R\$ 80,00

Exibindo 20 resultados Linha 5 selecionada

Detalhes

ID: 5

Cliente: Julio Victor Gomes

Descrição: Venda de Produtos

Data de Pagamento: 16/08/2022

Valor da Cobrança: R\$ 131,00

Valor Pago: R\$ 131,00

Fonte: Elaboração própria (2022).

Partindo agora para o item de menu “Vendas”, a Figura 48 apresenta a tela de gerenciamento de vendas. Do mesmo modo que as anteriores, ela é composta por uma tabela de visualização, preenchida com os registros das vendas que retornam do banco de dados; um painel de detalhes, que contém os atributos da entidade *Sale* em sua totalidade; e nesse caso, quatro controles de interação (dois para o mecanismo de pesquisa e mais dois botões).

Figura 48 – Tela de gerenciamento de vendas

Vendas

BoxGym Administrativo Financeiro Vendas Treinos Avaliações Ferramentas

Pesquisar... + Cadastrar Cancelar

Cliente	Data de Venda	Total	Status
Julio Victor Gomes	10/08/2022	R\$ 249,18	Pago
Felipe Anthony Novaes	11/08/2022	R\$ 105,00	Pago
Livia Nicole Antonella Galvão	12/08/2022	R\$ 45,00	Pago
Maya Bruna Milena da Cunha	15/08/2022	R\$ 49,80	Pago
Julio Victor Gomes	16/08/2022	R\$ 131,00	Pago
Lucas Oliveira Moura	18/08/2022	R\$ 45,00	Pago
Sabrina Barbosa da Silva	30/08/2022	R\$ 45,00	Pago
Danilo Fernandes Conceição	01/09/2022	R\$ 29,99	Pago
Marcos Azevedo Vieira	07/09/2022	R\$ 35,00	Pago
Danilo Fernandes Conceição	20/09/2022	R\$ 135,00	Pago
Livia Nicole Antonella Galvão	05/10/2022	R\$ 270,00	Pago
Pedro Lucas Cavalcanti	10/10/2022	R\$ 119,98	Pago
Lucas Oliveira Moura	26/10/2022	R\$ 196,50	Pago
Julio Victor Gomes	01/11/2022	R\$ 249,60	Pago
Marcos Azevedo Vieira	16/11/2022	R\$ 131,00	Aguardando pagamento
Livia Nicole Antonella Galvão	21/11/2022	R\$ 59,99	Aguardando pagamento

Exibindo 17 resultados Linha 1 selecionada

Detalhes

ID: 1
 Cliente: Julio Victor Gomes
 Data de Venda: 10/08/2022
 Total: R\$ 249,18
 Status: Pago
 Criação: 10/08/2022 20:49:06
 Modificação: 06/11/2022 21:09:59

Listar Produtos

Fonte: Elaboração própria (2022).

Ao selecionar uma linha da tabela, é possível saber quais produtos foram inseridos naquela venda específica clicando no botão “Listar Produtos”, localizado no painel de detalhes. Assim, uma janela como a da Figura 49 é aberta.

Figura 49 – Tela de visualização de itens de venda

Produto	Quantidade	Preço Unitário	Subtotal
Coqueteleira Espartanos Shaker 600 ml - Preto ...	2	R\$ 16,90	R\$ 33,80
Camiseta Goned Dry Touch Workout Feminina ...	2	R\$ 24,99	R\$ 49,98
Creatina Black Skull Turbo 300 g	1	R\$ 65,50	R\$ 65,50
ISO X Protein 100% Pure Isolate 1800 g - Espart...	1	R\$ 99,90	R\$ 99,90

Exibindo 4 produtos

Total

R\$ 249,18

Fonte: Elaboração própria (2022).

Ao clicar no botão “Cadastrar”, a janela da Figura 50 pode ser vista. Para realizar uma venda, o usuário do sistema deve obrigatoriamente informar um cliente e a data em que a venda ocorreu. Após isso, deve adicionar os produtos (itens de venda) e suas quantidades. O botão “Salvar” tem a função de cadastrar a venda, os itens de venda e a cobrança gerada no banco de dados.

Figura 50 – Tela de cadastro de vendas

Cadastrar venda

Cliente * Data de Venda * +

Venda Produto * Quantidade * Preço Unitário * +

Produto	Qtde.	Preço Unitário	Subtotal
Não há conteúdo na tabela			

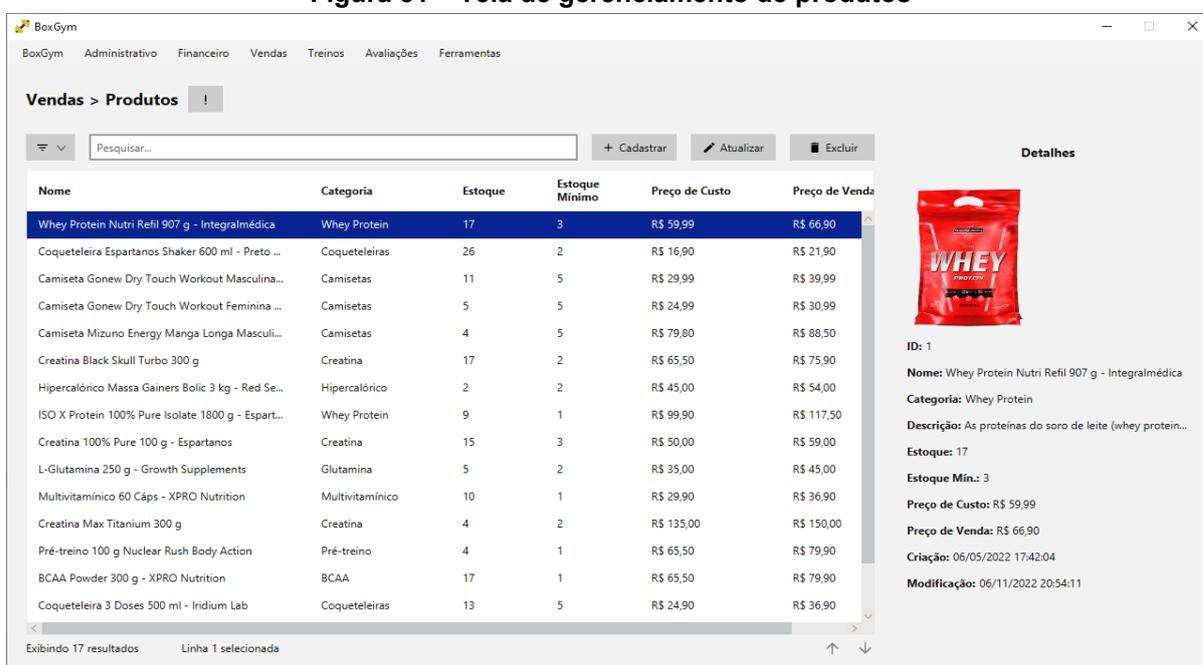
Exibindo nenhum produto

Total

Fonte: Elaboração própria (2022).

A Figura 51 apresenta a tela de gerenciamento de produtos. Semelhante às anteriores, essa tela é formada por uma tabela de visualização, preenchida com os registros dos produtos que retornam do banco de dados; um painel de detalhes, que contém os atributos da entidade *Product* em sua totalidade; e cinco controles de interação. O botão com o ícone do ponto de exclamação à direita do título da página, quando está ativo, indica que há produtos com a quantidade em estoque abaixo do mínimo, e quando pressionado, exibe um aviso que contém uma lista de quais produtos são esses.

Figura 51 – Tela de gerenciamento de produtos



Vendas > Produtos !

Pesquisar... + Cadastrar Atualizar Excluir

Nome	Categoria	Estoque	Estoque Mínimo	Preço de Custo	Preço de Venda
Whey Protein Nutri Refil 907 g - Integralmédica	Whey Protein	17	3	R\$ 59,99	R\$ 66,90
Coqueteleira Espartanos Shaker 600 ml - Preto ...	Coqueteleiras	26	2	R\$ 16,90	R\$ 21,90
Camiseta Goned Dry Touch Workout Masculina...	Camisetas	11	5	R\$ 29,99	R\$ 39,99
Camiseta Goned Dry Touch Workout Feminina ...	Camisetas	5	5	R\$ 24,99	R\$ 30,99
Camiseta Mizuno Energy Manga Longa Masculi...	Camisetas	4	5	R\$ 79,80	R\$ 88,50
Creatina Black Skull Turbo 300 g	Creatina	17	2	R\$ 65,50	R\$ 75,90
Hiperclórico Massa Gainers Bolic 3 kg - Red Se...	Hiperclórico	2	2	R\$ 45,00	R\$ 54,00
ISO X Protein 100% Pure Isolate 1800 g - Spart...	Whey Protein	9	1	R\$ 99,90	R\$ 117,50
Creatina 100% Pure 100 g - Espartanos	Creatina	15	3	R\$ 50,00	R\$ 59,00
L-Glutamina 250 g - Growth Supplements	Glutamina	5	2	R\$ 35,00	R\$ 45,00
Multivitamínico 60 Cáps - XPRO Nutrition	Multivitamínico	10	1	R\$ 29,90	R\$ 36,90
Creatina Max Titanium 300 g	Creatina	4	2	R\$ 135,00	R\$ 150,00
Pré-treino 100 g Nuclear Rush Body Action	Pré-treino	4	1	R\$ 65,50	R\$ 79,90
BCAA Powder 300 g - XPRO Nutrition	BCAA	17	1	R\$ 65,50	R\$ 79,90
Coqueteleira 3 Doses 500 ml - Iridium Lab	Coqueteleiras	13	5	R\$ 24,90	R\$ 36,90

Exibindo 17 resultados Linha 1 selecionada

Detalhes



ID: 1
Nome: Whey Protein Nutri Refil 907 g - Integralmédica
Categoria: Whey Protein
Descrição: As proteínas do soro de leite (whey protein...
Estoque: 17
Estoque Mín: 3
Preço de Custo: R\$ 59,99
Preço de Venda: R\$ 66,90
Criação: 06/05/2022 17:42:04
Modificação: 06/11/2022 20:54:11

Fonte: Elaboração própria (2022).

Ao clicar no botão “Cadastrar”, a tela da Figura 52 é aberta. Nela, o usuário deve informar o nome, estoque inicial, estoque mínimo, preço de custo e preço de venda do produto a ser cadastrado (os demais campos são opcionais).

Figura 52 – Tela de cadastro de produtos

Cadastrar produto

Nome * Categoria

Descrição

Estoque Inicial * Estoque Mínimo * Preço de Custo * Preço de Venda *

PRODUTO SEM IMAGEM

Salvar Limpar

Fonte: Elaboração própria (2022).

Caso seja necessário atualizar o cadastro de algum produto, o usuário do sistema deve selecioná-lo na tabela de visualização e clicar no botão “Atualizar”. Ao fazer isso, uma janela é aberta com todos os dados previamente cadastrados daquele produto para que possam ser modificados, conforme a Figura 53.

Figura 53 – Tela de atualização de produtos

Atualizar produto

Nome * Categoria

Descrição

Estoque Atual Estoque Mínimo * Preço de Custo Preço de Venda

WHEY PROTEIN

Salvar Limpar

Fonte: Elaboração própria (2022).

A Figura 54 apresenta a tela de gerenciamento de entradas de estoque. Do mesmo modo que as anteriores, ela é constituída por uma tabela de visualização, preenchida com os registros das entradas de estoque que retornam do banco de dados; um painel de detalhes, que contém os atributos da entidade *StockEntry* em sua totalidade; e quatro controles de interação.

Figura 54 – Tela de gerenciamento de entradas de estoque

The screenshot shows the 'BoxGym' application interface. The main menu includes 'Administrativo', 'Financeiro', 'Vendas', 'Treinos', 'Avaliações', and 'Ferramentas'. The current view is 'Vendas > Entradas de Estoque'. At the top, there is a search bar labeled 'Pesquisar...' and two buttons: '+ Cadastrar' and 'Cancelar'. Below this is a table with the following columns: 'Fornecedor', 'Data de Emissão', 'Nota Fiscal', and 'Total'. The table contains 17 rows of data. The first row is highlighted in blue. To the right of the table is a 'Detalhes' panel with the following information: 'ID: 1', 'Fornecedor: Ns2.com Internet S.A.', 'Data de Emissão: 01/09/2022', 'Nota Fiscal: 4009348', 'Total: R\$ 698,87', 'Criação: 01/09/2022 19:24:24', and 'Modificação: 01/09/2022 19:24:24'. At the bottom of the details panel is a button labeled 'Listar Produtos'. At the bottom of the table, it says 'Exibindo 17 resultados' and 'Linha 1 selecionada'.

Fornecedor	Data de Emissão	Nota Fiscal	Total
Ns2.com Internet S.A.	01/09/2022	4009348	R\$ 698,87
SBF Comércio de Produtos Esportivos LTDA	08/09/2022	4426871	R\$ 356,10
Suplemento Curitiba Afjr LTDA	15/09/2022	4010325	R\$ 49,80
Brg Suplementos Nutricionais LTDA	22/09/2022	4033658	R\$ 100,00
Ns2.com Internet S.A.	29/09/2022	4451474	R\$ 59,98
Ns2.com Internet S.A.	06/10/2022	4545121	R\$ 59,80
Ns2.com Internet S.A.	13/10/2022	4357583	R\$ 150,00
Ns2.com Internet S.A.	20/10/2022		R\$ 201,00
Growth Supplements - Produtos Alimentícios Eireli	27/10/2022		R\$ 250,00
Growth Supplements - Produtos Alimentícios Eireli	03/11/2022		R\$ 101,40
Growth Supplements - Produtos Alimentícios Eireli	10/11/2022		R\$ 154,95
Ns2.com Internet S.A.	14/11/2022		R\$ 90,00
Growth Supplements - Produtos Alimentícios Eireli	16/11/2022	4879520	R\$ 155,40
BT1 Suplementos Alimentares LTDA	18/11/2022	4008521	R\$ 327,50
SBF Comércio de Produtos Esportivos LTDA	21/11/2022	4005100	R\$ 131,00
SBF Comércio de Produtos Esportivos LTDA	24/11/2022	4002111	R\$ 205,70

Fonte: Elaboração própria (2022).

Ao selecionar uma linha da tabela, é possível saber quais produtos foram inseridos naquela entrada de estoque específica, para isso basta clicar no botão “Listar Produtos” localizado no painel de detalhes. Assim, uma janela como a da Figura 55 é aberta.

Figura 55 – Tela de visualização de itens de estoque

Produto	Quantidade	Preço de Custo	Subtotal
Whey Protein Nutri Refil 907 g - Integralmédica	3	R\$ 59,99	R\$ 179,97
Coqueteleira Espartanos Shaker 600 ml - Preto ...	6	R\$ 16,90	R\$ 101,40
Hiperclórico Massa Gainers Bolic 3 kg - Red Se...	2	R\$ 45,00	R\$ 90,00
Creatina Black Skull Turbo 300 g	5	R\$ 65,50	R\$ 327,50

Exibindo 4 produtos

Total

R\$ 698,87

Fonte: Elaboração própria (2022).

Ao clicar no botão “Cadastrar”, a janela da Figura 56 pode ser vista. Para realizar uma entrada de estoque, o usuário do sistema deve obrigatoriamente informar um fornecedor e a data de emissão da nota fiscal. Depois disso, deve adicionar os produtos (itens da entrada de estoque) e suas quantidades.

Figura 56 – Tela de cadastro de entradas de estoque

Fornecedor *
 Seleção

Data de Emissão *
 Escolha uma data

Nota Fiscal

Entrada de Estoque

Produto *
 Seleção

Quantidade *
 Preço de Custo *
 R\$ 0,00

Produto	Qtde.	Preço de Custo	Subtotal
Não há conteúdo na tabela			

Exibindo nenhum produto

Total

R\$ 0,00

Salvar Limpar

Fonte: Elaboração própria (2022).

O item de menu “Treinos” é o próximo a ser descrito neste projeto, começando, dessa vez, pelo seu último subitem, o qual é responsável pelo gerenciamento dos

exercícios executados na academia. Nessa tela, conforme a Figura 57, além dos componentes já vistos anteriormente, há ainda um botão com ícone de uma seta apontada para baixo (símbolo de *download*) à direita do título da página, que quando está ativo, permite importar dezenas de exercícios pré-cadastrados automaticamente.

Figura 57 – Tela de gerenciamento de exercícios

The screenshot shows the 'Treinos > Exercícios' management screen in the BoxGym application. The interface includes a search bar, buttons for '+ Cadastrar', 'Atualizar', and 'Excluir', and a table of exercises. The table has columns for 'Nome', 'Tipo', and 'Grupo'. The first row is highlighted, showing 'Abdominal Amplitude Máxima' as a 'Funcional' exercise for the 'Abdômen' group. To the right, a 'Detalhes' panel displays information for the selected exercise, including its ID, name, type, group, description, instructions, creation date, and modification date.

Nome	Tipo	Grupo
Abdominal Amplitude Máxima	Funcional	Abdômen
Abdominal Bicicleta	Funcional	Abdômen
Abdominal Canivete	Funcional	Abdômen
Abdominal Canoa	Funcional	Abdômen
Abdominal Giro Russo	Funcional	Abdômen
Extensão de Punho	Musculação	Antebraço
Rosca Inversa no Cross	Musculação	Antebraço
Rosca Alternada	Musculação	Bíceps
Rosca Concentrada	Musculação	Bíceps
Rosca Martelo Inclinado	Musculação	Bíceps
Agachamento com Elevação Frontal	Funcional	Corpo
Corda Naval Serpente	Crossfit	Corpo
Extensão Lombar	Musculação	Costas
Puxada na Barra Fixa	Musculação	Costas
Remada Alta Hammer	Musculação	Costas
Remada na Polia	Musculação	Costas

Detalhes

ID: 1
Nome: Abdominal Amplitude Máxima
Tipo: Funcional
Grupo: Abdômen
Descrição: Exercício com foco no fortalecimento e hip...
Instrução: 1. Deite sobre um colchonete; 2. Manter os...
Criação: 09/11/2022 17:37:43
Modificação: 09/11/2022 17:37:43

Exibindo 70 resultados Linha 1 selecionada

Fonte: Elaboração própria (2022).

Caso o usuário queira cadastrar um exercício que não está incluso nessa lista predefinida, basta clicar no botão “Cadastrar” para visualizar a tela da Figura 58, na qual deve ser informado, obrigatoriamente, o nome do exercício, o tipo (aeróbico, alongamento, funcional, musculação, entre outros) e principal parte do corpo exercitada (abdômen, antebraço, bíceps, costas, tríceps, etc.).

Figura 58 – Tela de cadastro de exercícios

A janela 'Cadastrar exercício' apresenta os seguintes campos e controles:

- Nome *: Campo de texto vazio.
- Tipo *: Menu suspenso com o valor 'Selecione'.
- Grupo *: Menu suspenso com o valor 'Selecione'.
- Descrição: Área de texto grande vazia.
- Instrução: Área de texto grande vazia.
- Botões: 'Salvar' e 'Limpar'.

Fonte: Elaboração própria (2022).

Caso seja necessário atualizar o cadastro de algum exercício, o usuário do sistema deve selecioná-lo na tabela de visualização e clicar no botão “Atualizar”. Ao fazer isso, uma janela como a da Figura 59 é aberta com todos os dados previamente cadastrados daquele exercício para que possam ser modificados.

Figura 59 – Tela de atualização de exercícios

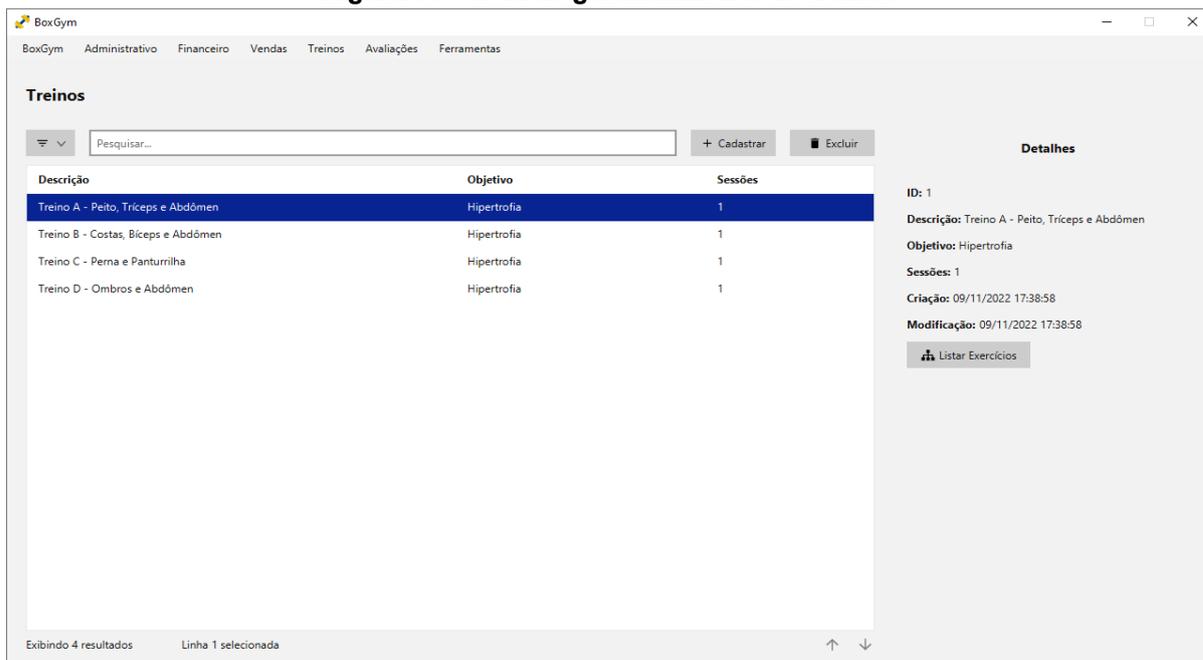
A janela 'Atualizar exercício' apresenta os seguintes campos e controles:

- Nome *: Campo de texto contendo 'Abdominal Amplitude Máxima'.
- Tipo *: Menu suspenso com o valor 'Funcional'.
- Grupo *: Menu suspenso com o valor 'Abdômen'.
- Descrição: Área de texto contendo 'Exercício com foco no fortalecimento e hipertrofia da região abdominal.'
- Instrução: Área de texto contendo:
 1. Deite sobre um colchonete;
 2. Manter os joelhos flexionados, pés ligeiramente separados;
 3. Coloque as mãos entrelaçadas atrás do pescoço;
 4. Inspire e eleve os ombros em direção ao joelho;
 5. Expire lentamente e retorne à posição inicial.
- Botões: 'Salvar' e 'Limpar'.

Fonte: Elaboração própria (2022).

A Figura 60 ilustra a tela de gerenciamento de treinos. Da mesma maneira que as outras, ela é constituída por uma tabela de visualização, preenchida com os registros dos treinos que retornam do banco de dados; um painel de detalhes, que contém os atributos da entidade *Workout* em sua totalidade; e quatro controles de interação.

Figura 60 – Tela de gerenciamento de treinos



The screenshot displays the 'Treinos' management interface in the BoxGym application. The interface includes a navigation menu with options like 'Administrativo', 'Financeiro', 'Vendas', 'Treinos', 'Avaliações', and 'Ferramentas'. Below the menu, there is a search bar and buttons for '+ Cadastrar' and 'Excluir'. The main area contains a table with the following data:

Descrição	Objetivo	Sessões
Treino A - Peito, Tríceps e Abdômen	Hipertrofia	1
Treino B - Costas, Biceps e Abdômen	Hipertrofia	1
Treino C - Perna e Panturrilha	Hipertrofia	1
Treino D - Ombros e Abdômen	Hipertrofia	1

To the right of the table is a 'Detalhes' panel for the selected workout (Treino A). It displays the following information:

- ID: 1
- Descrição: Treino A - Peito, Tríceps e Abdômen
- Objetivo: Hipertrofia
- Sessões: 1
- Criação: 09/11/2022 17:38:58
- Modificação: 09/11/2022 17:38:58

At the bottom of the details panel, there is a button labeled 'Listar Exercícios'. The status bar at the bottom of the application indicates 'Exibindo 4 resultados' and 'Linha 1 selecionada'.

Fonte: Elaboração própria (2022).

Ao selecionar uma linha da tabela, é possível saber quais exercícios foram inseridos naquele treino específico, para isso basta clicar no botão “Listar Exercícios” localizado no painel de detalhes. Assim, uma janela semelhante à da Figura 61 se abre.

Figura 61 – Tela de visualização dos exercícios contidos nos treinos

Nome do Exercício	Séries	Repetições	Descanso (em segundos)
Supino Inclinado com Halteres	4	8	60
Supino Reto com Barra	4	8	60
Supino Vertical na Máquina	4	8	60
Crucifixo Declinado com Halteres	4	8	60
Tríceps Corda no Cross Over	4	8	60
Tríceps Coice na Polia	4	8	60
Tríceps Francês	4	8	60
Abdominal Infra	4	15	60

Exibindo 8 exercícios

Fonte: Elaboração própria (2022).

A tela de cadastro de treinos pode ser vista na Figura 62. Para inserir um treino, o usuário do sistema deve preencher alguns campos obrigatórios, dentre eles: descrição, objetivo (condicionamento, emagrecimento, hipertrofia ou reabilitação física), quantidade de sessões a serem realizadas e inserir um ou mais exercícios. Opcionalmente, é possível informar a quantidade de séries, repetições e tempo de descanso para cada exercício.

Figura 62 – Tela de cadastro de treinos

Descrição *
 Objetivo * Selecione
 Sessões *
 +

Treino
 Grupo * Selecione
 Nome do Exercício * ---
 Séries
 Repetições
 Descanso
 +

Nome do Exercício	Séries	Repetições	Descanso (em segundos)
Não há conteúdo na tabela			

Exibindo nenhum exercício

Fonte: Elaboração própria (2022).

Na tela de gerenciamento das fichas dos clientes, conforme a Figura 63, além dos componentes já apresentados, há também um botão com ícone de atualização, que quando está ativo, sinaliza a existência de fichas vencidas que podem ser desativadas. Dessa forma, ao pressioná-lo, o atributo *status* dessas fichas é alterado e elas deixam de aparecer na tabela.

Figura 63 – Tela de gerenciamento de fichas

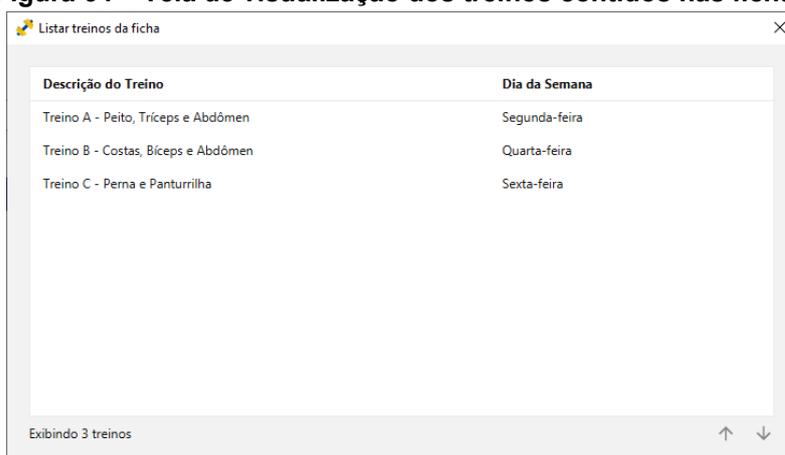
The screenshot displays the 'BoxGym' application interface. At the top, there is a navigation menu with options: BoxGym, Administrativo, Financeiro, Vendas, Treinos, Avaliações, and Ferramentas. Below the menu, the breadcrumb 'Treinos > Fichas' is visible. A search bar labeled 'Pesquisar...' is present, along with '+ Cadastrar' and 'Excluir' buttons. The main area contains a table with the following data:

Cliente	Descrição da Ficha	Data de Validade	Observações
Julio Victor Gomes	Ficha 1	01/12/2022	
Livia Nicole Antonella Galvão	Ficha 1	10/12/2022	

To the right of the table is a 'Detalhes' panel for the selected row (Julio Victor Gomes). It shows: ID: 1, Cliente: Julio Victor Gomes, Descrição da Ficha: Ficha 1, Data de Validade: 01/12/2022, Observações: (empty), Criação: 09/11/2022 17:46:53, and Modificação: 09/11/2022 17:46:53. A 'Listar Treinos' button is located at the bottom of the details panel. At the bottom of the interface, it says 'Exibindo 2 resultados' and 'Linha 1 selecionada'.

Fonte: Elaboração própria (2022).

Para visualizar os treinos inseridos em uma determinada ficha, tal qual o exemplo da Figura 64, é preciso selecioná-la na tabela e pressionar o botão “Listar Treinos” presente no painel de detalhes.

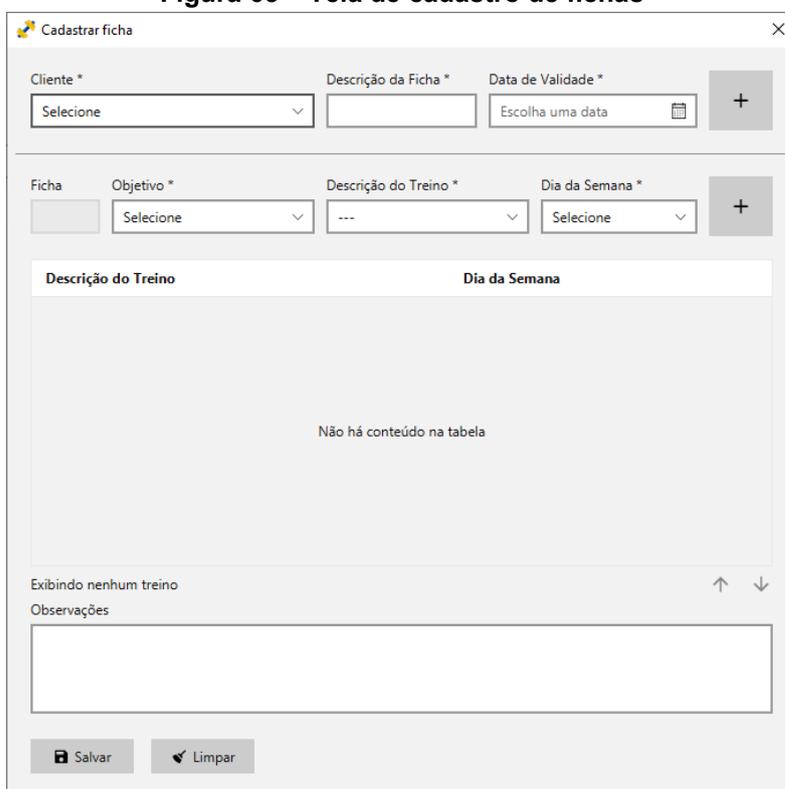
Figura 64 – Tela de visualização dos treinos contidos nas fichas


Descrição do Treino	Dia da Semana
Treino A - Peito, Tríceps e Abdômen	Segunda-feira
Treino B - Costas, Bíceps e Abdômen	Quarta-feira
Treino C - Perna e Panturrilha	Sexta-feira

Exibindo 3 treinos

Fonte: Elaboração própria (2022).

A tela de cadastro de ficha pode ser vista na Figura 65. Para inserir uma ficha no sistema, o usuário deve preencher os seguintes campos obrigatórios: nome do cliente, descrição e data de validade, e inserir um ou mais treinos, selecionando também em qual dia da semana cada um deve ser feito. O campo dedicado às observações é opcional.

Figura 65 – Tela de cadastro de fichas


Cadastrar ficha

Cliente *
 Descrição da Ficha *
 Data de Validade *

Ficha
 Objetivo *
 Descrição do Treino *
 Dia da Semana *

Descrição do Treino	Dia da Semana
Não há conteúdo na tabela	

Exibindo nenhum treino

Observações

Fonte: Elaboração própria (2022).

Quanto ao único subitem de “Avaliações”, a tela de gerenciamento de medidas dos perímetros corporais pode ser visualizada na Figura 66. Semelhante às outras, ela é constituída por uma tabela de visualização, preenchida com os registros das medidas que retornam do banco de dados; um painel de detalhes, que contém os atributos da entidade *Measurement* em sua totalidade; e cinco controles de interação.

Figura 66 – Tela de gerenciamento de medidas

Cliente	Data de Medição	Altura (em cm)	Peso (em kg)
Julio Victor Gomes	07/11/2022	180	78,0
Felipe Anthony Novaes	08/11/2022	175	72,0
Lucas Oliveira Moura	09/11/2022	193	71,0
Luiza Antônia de Paula	09/11/2022	160	54,0

Detalhes

ID: 1
 Cliente: Julio Victor Gomes
 Data de Medição: 07/11/2022
 Altura: 180 cm
 Peso: 78,0 kg
 Pescoço: 39,7 cm
 Ombro: 0,0 cm
 Braço Dir.: 37,2 cm Braço Esq.: 37,2 cm
 Antebraço Dir.: 31,1 cm Antebraço Esq.: 31,1 cm
 Tórax: 103,8 cm
 Cintura: 76,6 cm
 Abdômen: 0,0 cm
 Quadril: 93,3 cm
 Coxa Dir.: 56,1 cm Coxa Esq.: 56,1 cm
 Panturrilha Dir.: 37,2 cm Panturrilha Esq.: 37,2 cm
 Criação: 06/11/2022 17:27:33
 Modificação: 06/11/2022 17:27:33

Exibindo 4 resultados Linha 1 selecionada

Fonte: Elaboração própria (2022).

Para cadastrar as medidas de um cliente, o usuário do sistema deve selecioná-lo e informar sua altura e peso, além de preencher o campo com a data em que a medição está sendo feita. Com o objetivo de auxiliar no preenchimento das medidas, essa tela conta com a imagem da silhueta do corpo humano, que varia conforme o sexo do cliente selecionado para indicar a localização dos segmentos corporais. A tela de cadastro de medidas pode ser vista logo abaixo na Figura 67.

Figura 67 – Tela de cadastro de medidas

Cadastrar medidas

Cliente * Sexo Idade Data * Altura * Peso *

Pescoço

Ombro

Braço Direito Braço Esquerdo

Antebraço Direito Antebraço Esquerdo

Tórax

Cintura

Abdômen

Quadril

Coxa Direita Coxa Esquerda

Panturrilha Direita Panturrilha Esquerda

Fonte: Elaboração própria (2022).

Caso seja preciso atualizar as medidas de algum cliente, pode-se fazer essa ação na tela de atualização de medidas, conforme a Figura 68.

Figura 68 – Tela de atualização de medidas

Atualizar medidas

Cliente * Sexo Idade Data * Altura * Peso *

Pescoço

Ombro

Braço Direito Braço Esquerdo

Antebraço Direito Antebraço Esquerdo

Tórax

Cintura

Abdômen

Quadril

Coxa Direita Coxa Esquerda

Panturrilha Direita Panturrilha Esquerda

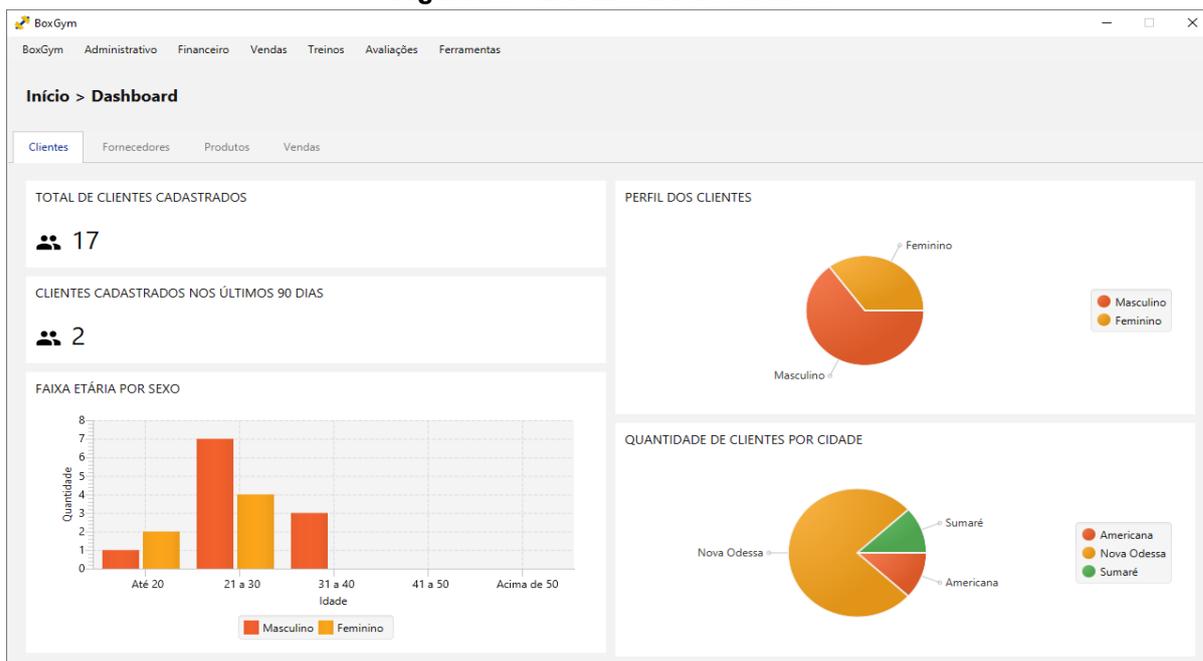
Fonte: Elaboração própria (2022).

Por fim, para finalizar a apresentação das telas deste sistema de academia, o subitem “Início”, primeiro subitem que o usuário visualiza após efetuar o *login* e que está localizado no item de menu “BoxGym”, será comentado a seguir.

Ele contém um controle do JavaFX que permite alternar entre um grupo de guias (apenas uma guia é visível por vez). Cada guia contém um determinado número de painéis que, juntos, compõem uma *dashboard* dedicada a um módulo do sistema. No total, são quatro *dashboards* responsáveis por fornecer visualizações rápidas e intuitivas referentes aos clientes, fornecedores, produtos e vendas da academia. Para tal, o sistema executa consultas SQL nas tabelas do banco de dados e utiliza os resultados retornados para preencher os rótulos de texto ou gráficos existentes nos painéis.

A primeira guia contém a *dashboard* de clientes, conforme a Figura 69.

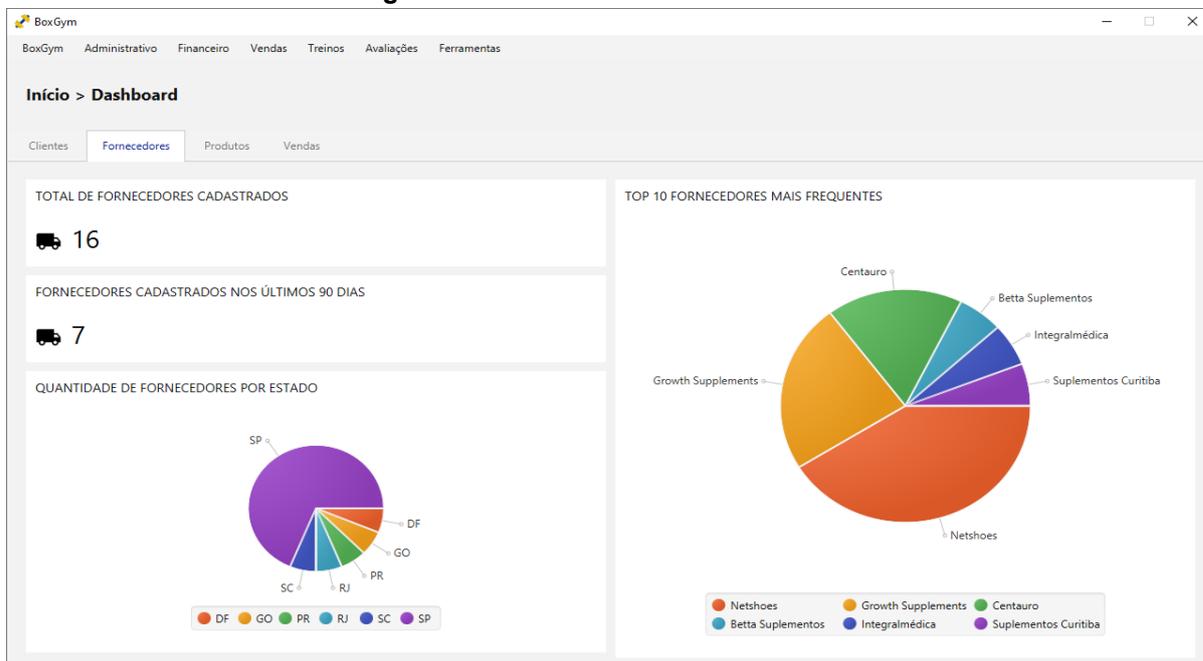
Figura 69 – Dashboard de clientes



Fonte: Elaboração própria (2022).

A segunda guia contém a *dashboard* de fornecedores, conforme a Figura 70.

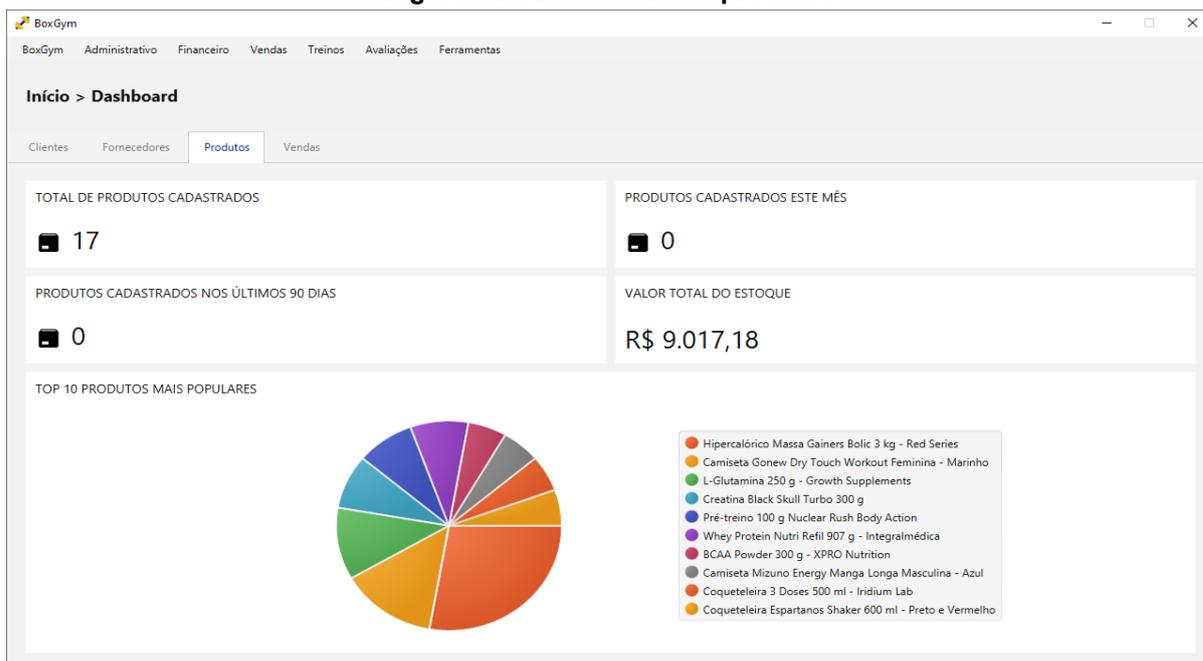
Figura 70 – Dashboard de fornecedores



Fonte: Elaboração própria (2022).

A terceira guia contém a *dashboard* de produtos, conforme a Figura 71.

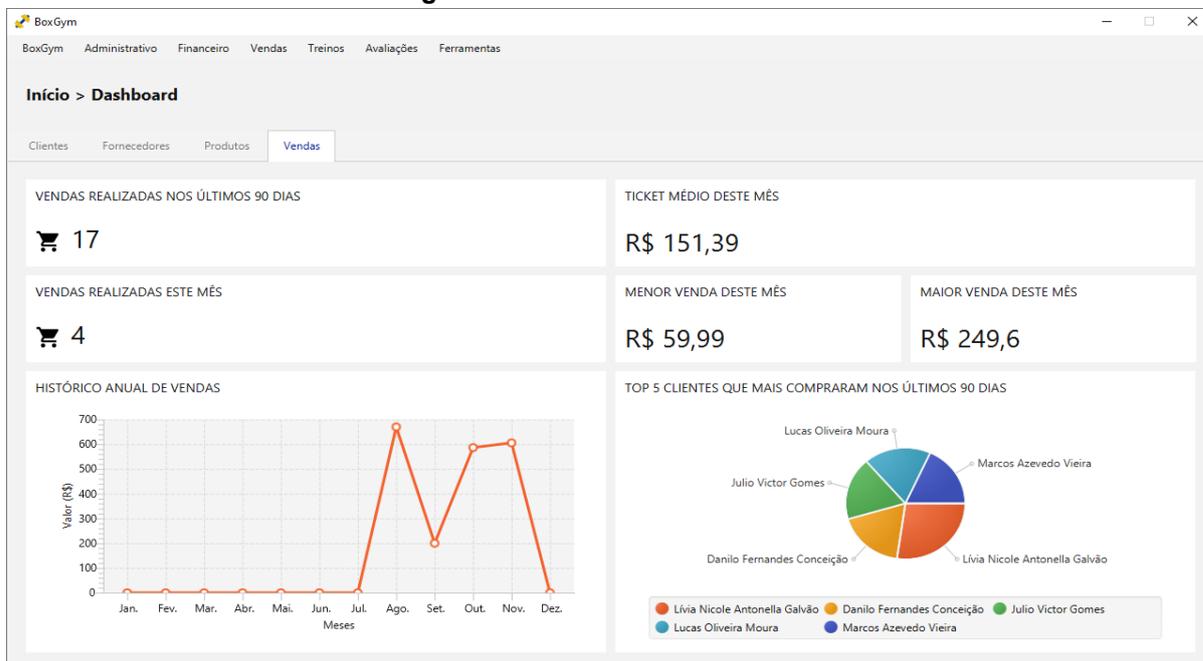
Figura 71 – Dashboard de produtos



Fonte: Elaboração própria (2022).

A quarta guia contém a *dashboard* de vendas, conforme a Figura 72.

Figura 72 – Dashboard de vendas



Fonte: Elaboração própria (2022).

5 CONSIDERAÇÕES FINAIS

Utilizando como tema uma pequena academia de musculação e treinamento funcional situada em Nova Odessa (SP) chamada BoxGym, na qual o investimento em tecnologia nunca foi uma prioridade de seu proprietário, este trabalho acadêmico teve o objetivo de projetar e desenvolver um sistema *desktop* de gerenciamento para auxiliá-lo, principalmente em suas dificuldades relacionadas à gestão de clientes, produtos e pagamentos.

Para se atingir esse objetivo, foi preciso realizar uma análise prévia dos sistemas concorrentes existentes no mercado que apresentavam um propósito semelhante, levantar e registrar os requisitos funcionais e não funcionais, assimilar os recursos e ferramentas escolhidos, especialmente a linguagem de programação Java e a plataforma de aplicação JavaFX, e fazer a modelagem do sistema com o uso de diagramas da UML. Apoiado nos conceitos descritos nas metodologias ágeis, especificamente na metodologia *Scrum*, o trabalho foi fracionado em 11 ciclos de desenvolvimento com duração de 14 dias (com exceção da terceira *Sprint*), nos quais, ao final de cada um, pelo menos um requisito funcional era implementado, isto é, uma nova funcionalidade principal era adicionada ao sistema em um processo contínuo. Como resultado, obteve-se um sistema de computador que atendeu todos os requisitos da BoxGym.

Durante o andamento do projeto, muitas adversidades foram encontradas no desenvolvimento do código-fonte, que constantemente exigia a leitura de diversas e longas páginas de documentação ou de fóruns na Internet dedicados a programadores para solucionar os eventuais erros que eram identificados na execução do sistema. Além disso, um outro obstáculo marcante enfrentado pela equipe nesse decurso, sobretudo nos primeiros meses, foi o tópico da modelagem do sistema em razão da complexidade e quantidade de funções que deveriam ser projetadas.

Apesar disso, diante de todos os problemas e com foco em solucioná-los, os integrantes da equipe aplicaram, de maneira prática, inúmeros conhecimentos e fundamentos que, muitas vezes, tinham sido vistos apenas de forma teórica ou em cenários mais básicos nas disciplinas ministradas no curso de Análise e Desenvolvimento de Sistemas, proporcionando assim, experiências e novas

habilidades de desenvolvimento e trabalho em equipe valiosas e muito importantes, por exemplo, para o mercado de trabalho.

Mesmo que todos os requisitos funcionais tenham sido desenvolvidos com êxito, ainda há espaço para expandir as funcionalidades do sistema ou complementar as existentes. Uma adição que pode ser feita em versões posteriores é a criação de um módulo dedicado ao gerenciamento das contas a pagar, no qual o usuário seria capaz de organizar as descrições, prazos e valores de todas as obrigações financeiras assumidas pela academia, ou seja, qualquer tipo de pagamento efetuado por ela em seu cotidiano, seja de colaboradores, fornecedores, aluguel ou até mesmo de contas de consumo, como de água, energia elétrica, entre outras.

Por fim, é desejável que este trabalho possa colaborar, em algum nível, com as próximas aplicações acadêmicas relacionadas ao tema. Com foco nas futuras contribuições de analistas e desenvolvedores, sugere-se o desenvolvimento de um sistema voltado ao ambiente *web* ou *mobile*, dada a relevância e popularidade dessas plataformas atualmente, ou a criação de mais uma interface além da utilizada pelo gestor da academia, para que clientes e alunos também possam acompanhar os assuntos pertinentes a eles, como informações cadastrais, treinos atribuídos, histórico de compras e pagamentos e evolução corporal de cada um.

REFERÊNCIAS

AGILE ALLIANCE. Agile Manifesto. **Agile Alliance**, 2019. Disponível em: <https://www.agilealliance.org/wp-content/uploads/2019/09/agile-manifesto-download-2019.pdf>. Acesso em: 11 maio 2022, às 14h38.

ANYDESK. **AnyDesk user manual for Windows, macOS and Linux**. [S. l.: s. n.], 2017. 53 p. Disponível em: <https://download.anydesk.com/docs/AnyDesk-UserManual.pdf>. Acesso em: 27 mar. 2022, às 10h13.

APACHE HTTP SERVER PROJECT. About the Apache HTTP Server Project. **APACHE HTTP SERVER PROJECT**, 1997. Disponível em: https://httpd.apache.org/ABOUT_APACHE.html. Acesso em: 25 mar. 2022, às 21h53.

ASTAH. Astah UML. **Astah**, 2022. Disponível em: <https://astah.net/products/astah-uml/>. Acesso em: 19 mar. 2022, às 14h20.

ATLASSIAN. Scrum: saiba como usar o Scrum da melhor forma. **Atlassian**, 2022. Disponível em: <https://www.atlassian.com/br/agile/scrum>. Acesso em: 11 maio 2022, às 15h58.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 2. ed. rev. e atual. Rio de Janeiro: Elsevier, 2007. ISBN 85-352-1696-0.

CORONEL, Carlos; MORRIS, Steven. **Database systems: design, implementation, and management**. 12th ed. Boston: Cengage Learning, 2016. ISBN 978-1-305-62748-2.

EDUCBA. What is XAMPP? **EDUCBA**, 2021. Disponível em: <https://www.educba.com/what-is-xampp/>. Acesso em: 25 mar. 2022, às 21h44.

GIT. Página inicial. **Git**, 2022. Disponível em: <https://git-scm.com/>. Acesso em: 23 mar. 2022, às 10h23.

GITHUB. Introdução ao GitHub Desktop. **GitHub**, 2022. Disponível em: <https://docs.github.com/pt/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop>. Acesso em: 27 mar. 2022, às 9h44.

GLOBAL WELLNESS INSTITUTE. The global wellness economy. *In*: GLOBAL WELLNESS INSTITUTE. **The global wellness economy: looking beyond COVID**. Miami: Global Wellness Institute, 2021. p. 1-18. Disponível em: https://globalwellnessinstitute.org/wp-content/uploads/2021/11/GWI-WE-Monitor-2021_final-digital.pdf. Acesso em: 12 maio. 2022, às 13h20.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. 2. ed. São Paulo: Novatec Editora, 2011. ISBN 978-85-7522-281-2.

HARTMAN, James. JDK vs JRE vs JVM: key differences. **Guru99**, 2022. Disponível em: <https://www.guru99.com/difference-between-jdk-jre-jvm.html#:~:text=JDK%20is%20a%20software%20development%20kit%20whereas%20JRE%20is%20a,JVM%20is%20Java%20Virtual%20Machine>. Acesso em: 23 mar. 2022, às 18h45.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009. *E-book* (282 p.). ISBN 978-85-7780-452-8.

IBM CLOUD EDUCATION. What is Java? **IBM**, 2019. Disponível em: <https://www.ibm.com/cloud/learn/java-explained>. Acesso em: 20 mar. 2022, às 16h03.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE standard glossary of software engineering terminology**. New York: IEEE Standards Board, 1990. 84 p. DOI 10.1109/IEEESTD.1990.101064. Disponível em: <https://ieeexplore.ieee.org/document/159342>. Acesso em: 25 abr. 2022, às 15h47.

INTERNATIONAL HEALTH, RACQUET & SPORTSCLUB ASSOCIATION. **The 2020 IHRSA Global Report: the state of the health club industry**. [S. l.]: IHRSA, 2020. Disponível em: <https://www.ihrsa.org/publications/the-2020-ihrsa-global-report/>. Acesso em: 9 maio 2022, às 18h40.

LOELIGER, Jon; MCCULLOUGH, Matthew. Introduction. *In*: LOELIGER, Jon; MCCULLOUGH, Matthew. **Version control with Git: powerful tools and techniques for collaborative software development**. 2nd ed. Sebastopol, CA: O'Reilly Media, 2012. p. 1-8 (cap. 1). ISBN 978-1-449-31638-9.

LOWE, Doug. Getting started with JavaFX. *In*: LOWE, Doug. **JavaFX for dummies**. New Jersey: John Wiley & Sons, 2014. p. 7-30 (pt. 1).

MARIADB FOUNDATION. Página inicial. **MariaDB Foundation - MariaDB.org**, 2022. Disponível em: <https://mariadb.org/>. Acesso em: 25 mar. 2022, às 21h42.

MELONI, Julie C. Understanding Cascading Style Sheets. *In*: MELONI, Julie C. **Sams teach yourself HTML, CSS, and JavaScript all in one**. Indianapolis: Pearson Education, 2011. p. 45-64 (cap. 3). ISBN 978-0-672-33332-3.

OPENJFX. Getting started with JavaFX. **OpenJFX**, 2022. Disponível em: <https://openjfx.io/openjfx-docs/>. Acesso em: 28 mar. 2022, às 12h22.

ORACLE. JavaFX CSS reference guide. **Oracle**, 2014a. Disponível em: <https://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html#intros scenegraph>. Acesso em: 26 mar. 2022, às 20h36.

ORACLE. JavaFX Scene Builder. **Oracle**, 2020. Disponível em: <https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html>. Acesso em: 20 mar. 2022, às 16h11.

ORACLE. NetBeans IDE. **Oracle**, 2018. Disponível em: <https://www.oracle.com/tools/technologies/netbeans-ide.html>. Acesso em: 20 mar. 2022, às 11h17.

ORACLE. O que é um banco de dados? **Oracle**, 2014b. Disponível em: <https://www.oracle.com/br/database/what-is-database/>. Acesso em: 28 abr. 2022, às 15h32.

ORACLE. The Java language environment. **Oracle**, 2019. Disponível em: <https://www.oracle.com/java/technologies/introduction-to-java.html>. Acesso em: 20 mar. 2022, às 16h03.

PHPMYADMIN. Página inicial. **phpMyAdmin**, 2022. Disponível em: <https://www.phpmyadmin.net/>. Acesso em: 25 mar. 2022, às 21h27.

PRESSMAN, Roger S. **Engenharia de software**: uma abordagem profissional. Tradução de Ariovaldo Griesi e Mario Moro Fecchio. 7. ed. Porto Alegre: AMGH, 2011. Título original: Software engineering: a practitioner's approach. ISBN 978-85-8055-044-3.

RISING, Linda; JANOFF, Norman S. The Scrum software development process for small teams. **IEEE Software**, [s. l.], v. 17, n. 4, p. 26-32, jul./aug. 2000. DOI 10.1109/52.854065. Disponível em: <https://ieeexplore.ieee.org/document/854065>. Acesso em: 20 abr. 2022, às 22h17.

ROBBINS, Jennifer Niederst. Introducing Cascading Style Sheets. *In*: ROBBINS, Jennifer Niederst. **Learning web design**: a beginner's guide to HTML, CSS, JavaScript, and web graphics. 5th ed. Sebastopol, CA: O'Reilly Media, 2018. p. 239-260 (cap. 11). ISBN 978-1-491-96020-2.

SCHWABER, Ken. Backdrop: the science of Scrum. *In*: SCHWABER, Ken. **Agile project management with Scrum**. Redmond: Microsoft Press, 2004. p. 1-14. ISBN 978-0-7356-1993-7. Disponível em: https://books.google.com.br/books?id=6pZCAwAAQBAJ&pg=PT29&source=gbs_selected_pages&cad=3#v=onepage&q&f=false. Acesso em: 26 mar. 2022, às 12h49.

SCHWABER, Ken. SCRUM Development Process. *In*: SUTHERLAND, J.; CASANAVE, C.; MILLER, J.; PATEL, P.; HOLLOWELL, G. (org.). **Business Object Design and Implementation**. London: Springer, 1997. p. 117-134. Disponível em: <http://jeffsutherland.com/oopsia/schwapub.pdf>. Acesso em: 20 abr. 2022, às 23h39.

SCHWABER, Ken; SUTHERLAND, Jeff. **O guia do Scrum**: o guia definitivo para o Scrum: as regras do jogo. Tradução de Fábio Cruz, Eduardo Rodrigues Sucena e Rodrigo Paulo. [S. l.]: ScrumGuides.org, 2020. *E-book* (16 p.). Título original: The Scrum guide — the definitive guide to Scrum: the rules of the game. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-2.0.pdf>. Acesso em: 26 mar. 2022, às 21h17.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. Tradução de Daniel Vieira. 5. ed. Rio de Janeiro: Elsevier, 2006. Título original: Database system concepts. ISBN 85-352-1107-8.

SILVERMAN, Richard E. Understanding Git. *In*: SILVERMAN, Richard E. **Git pocket guide**: a working introduction. Sebastopol, CA: O'Reilly Media, 2013. p. 1-32 (cap. 1). ISBN 978-1-449-32586-2.

SIS4.COM. brModelo 3.3. **Sis4.com**, 2020. Disponível em: <https://www.sis4.com/brModelo/>. Acesso em: 1 maio 2022, às 14h24.

SOMMERVILLE, Ian. **Engenharia de software**. Tradução de Ivan Bosnic e Kalinka Oliveira. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Título original: Software engineering. ISBN 978-85-7936-108-1.

SYDLE. Framework Scrum: o que é e como funciona? **Sydle**, 2020. Disponível em: <https://www.sydle.com/br/blog/framework-scrum-5f6dc45f320703787497f887/>. Acesso em: 23 maio 2022, às 07h50.

VIEIRA, Pedro Duque. JMetro – Java, JavaFX Theme. **Pixel Duke**, 2018. Disponível em: <https://www.pixelduke.com/java-javafx-theme-jmetro/>. Acesso em: 3 jun 2022, às 09h06.

VOS, Johan *et al.* Creating a user interface in JavaFX. *In*: VOS, Johan *et al.* **Pro JavaFX 9**: a definitive guide to building desktop, mobile, and embedded Java clients. 4th ed. New York: Apress, 2017. p. 33-82 (cap. 2). ISBN 978-1-4842-3042-8. Disponível em: <https://books.google.ca/books?id=eQIDDwAAQBAJ&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false>. Acesso em: 18 mar. 2022, às 15h15.

WIELENGA, Geertjan. Installing and setting up. *In*: WIELENGA, Geertjan. **Beginning NetBeans IDE**: for Java developers. New York: Apress, 2015. p. 3-16 (cap. 1). ISBN 978-1-4842-1257-8. Disponível em: https://books.google.fr/books?id=9HeBCgAAQBAJ&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false. Acesso em: 18 mar. 2022, às 10h08.