



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS ARARANGUÁ  
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE (CTS)  
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO (TIC)

Ana Vitória Barbosa Mariano

**Desenvolvimento de um Software de Gestão de Máquinas para  
Academia de Atividade Física**

Araranguá  
2023

Ana Vitória Barbosa Mariano

**Desenvolvimento de um Software de Gestão de Máquinas para Academia de  
Atividade Física**

Trabalho de Conclusão de Curso submetido ao curso de Tecnologias da Informação e Comunicação do Centro de Ciências, Tecnologias e Saúde da Universidade Federal de Santa Catarina como requisito parcial para obtenção do título de Bacharel em Tecnologias da Informação e Comunicação.

Orientadora: Profa. Dra. Eliane Pozzebon

Araranguá  
2023

Mariano, Ana Vitória Barbosa

Desenvolvimento de um Software de Gestão de Máquinas para Academia de Atividade Física / Ana Vitória Barbosa Mariano ; orientadora, Eliane Pozzebon, 2023.

119 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Tecnologias da Informação e Comunicação, Araranguá, 2023.

Inclui referências.

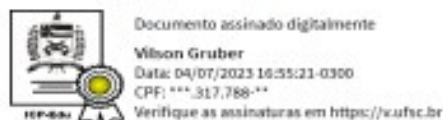
1. Tecnologias da Informação e Comunicação. 2. Software. 3. Gestão de Máquinas. 4. Academia. 5. Atividade Física. I. Pozzebon, Eliane. II. Universidade Federal de Santa Catarina. Graduação em Tecnologias da Informação e Comunicação. III. Título.

Ana Vitória Barbosa Mariano

## DESENVOLVIMENTO DE UM SOFTWARE DE GESTÃO DE MÁQUINAS PARA ACADEMIA DE ATIVIDADE FÍSICA

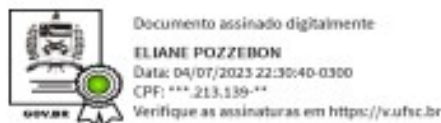
Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel e aprovado em sua forma final pelo Curso Tecnologias da Informação e Comunicação.

Araranguá, 30 de junho de 2023.



Prof. Wilson Gruber, Dr.  
Coordenador do Curso

### Banca Examinadora:



Prof.<sup>a</sup> Eliane Pozzebon, Dra.  
Orientadora

Universidade Federal de Santa Catarina



Prof.<sup>a</sup> Andréa Sabedra Bordin, Dra.  
Universidade Federal de Santa Catarina



Prof. Fabricio Herpich, Dr.  
Universidade Federal de Santa Catarina

Para o homem que me educou usando algoritmos:  
Roberval Mariano, meu pai.

## **AGRADECIMENTOS**

Primeiramente, gostaria de agradecer à Universidade Federal de Santa Catarina, que me capacitou tanto para este projeto quanto para o mercado de trabalho. Lá, tive professores dedicados e uma grade curricular de excelência, que contribuíram para o meu crescimento como profissional.

Gostaria também de agradecer à professora Eliane Pozzebon, minha orientadora, por todos os anos de ensino e orientação, não apenas no aspecto acadêmico, mas também na vida em geral.

Agradeço à minha mãe, Maria Conceição, por todo o apoio neste projeto, por revisar este documento e por me estimular a finalizá-lo.

Meu agradecimento também vai para meu companheiro de vida, Kelvin Gonçalves, que não apenas me apoiou na realização deste projeto, mas também me auxiliou a desenvolver as telas, me orientando para que toda a interface fosse agradável.

Agradeço ao meu amigo Edson Ney, que realizou todas as rodadas de teste do aplicativo e sempre se mostrou disponível para ajudar no que fosse necessário. Gostaria também de agradecer ao meu amigo Gabriel Moraes, que disponibilizou seu tempo para me auxiliar na implantação do sistema.

Aos professores Fabrício Herpich e Andrea Bordin, meu agradecimento por me ensinarem com dedicação os principais conteúdos necessários para a execução deste trabalho. Quero fazer um agradecimento especial à professora Marina Carradore, que, além de compartilhar conhecimentos, ajudou-me a coletar os dados finais deste projeto e se tornou uma amiga muito querida.

Expresso minha gratidão aos profissionais de Educação Física Aluísio, Bruna, Juliana, Patrícia e Renata, que se dispuseram a participar dos testes de usabilidade e qualidade: vocês foram fundamentais para o encerramento deste projeto com chave de ouro.

Agradeço, por fim, à minha terapeuta, Maria Regina, e à minha família, por me ajudarem a manter o foco neste projeto.

## RESUMO

Este trabalho tem como objetivo investigar as oportunidades de integração tecnológica no mercado de academias especializadas em treinamento personalizado. A proposta consiste no desenvolvimento de um software de gerenciamento de máquinas, por meio de uma aplicação web, que ofereça suporte às atividades dessas academias, proporcionando uma experiência aprimorada para os usuários. A implementação dessa solução tecnológica busca otimizar os processos de gerenciamento, visando um ambiente mais eficiente e adaptado às necessidades específicas das academias. A metodologia adotada neste estudo baseia-se em pesquisa de campo, aplicando os conhecimentos adquiridos durante o Curso de Tecnologia da Informação e da Comunicação. O método de desenvolvimento escolhido é a metodologia ágil SCRUM, que permite uma abordagem adaptativa e orientada por objetivos. A criação de documentos de especificação do software auxilia na definição dos objetivos estruturais, como a estrutura das conexões necessárias, telas e funcionalidades, bem como dos objetivos específicos, como os casos de uso e os fluxos de dados. A combinação da metodologia SCRUM com as práticas de engenharia de software proporciona um processo de desenvolvimento eficiente e orientado pela qualidade do software. Os resultados obtidos até o momento revelam uma excelente receptividade e aceitação do software de gerenciamento de máquinas para academias especializadas. Durante as entrevistas e a coleta de formulários, foi possível identificar que a maioria dos usuários expressou interesse em utilizar o software de forma frequente, demonstrando uma grande adaptabilidade do sistema ao público-alvo das academias especializadas. Além disso, os resultados apontam para a concordância dos usuários em relação à facilidade de uso, à integração das funções, à rapidez no aprendizado do sistema e à organização das informações. Esses aspectos são fundamentais para garantir a usabilidade e a satisfação dos usuários. Esses resultados positivos reforçam a importância da aplicação da tecnologia no contexto das academias especializadas em treinamento personalizado. A implementação do software de gerenciamento de máquinas possui o potencial de otimizar os processos de gerenciamento, oferecendo um ambiente mais eficiente e adaptado às necessidades das academias. Isso proporcionará uma experiência diferenciada para os usuários, contribuindo para a melhoria da eficiência operacional das academias e para a satisfação dos clientes.

**Palavras-chave:** Software; Metodologia Ágil; Academia; Gestão de Máquinas; Atividade Física.

## ABSTRACT

This work aims to investigate the opportunities for technological integration in the market of specialized gyms focused on personalized training. The proposal involves developing a machine management software through a web application that supports the activities of these gyms, providing an enhanced experience for users. The implementation of this technological solution aims to optimize management processes, creating a more efficient environment tailored to the specific needs of the gyms. The chosen development method is the agile SCRUM methodology, which allows for an adaptive and goal-oriented approach. Creating software specification documents assists in defining structural objectives, such as the required connections, screens, and functionalities, as well as specific goals, such as use cases and data flows. The combination of SCRUM methodology with software engineering practices provides an efficient development process focused on software quality. The results obtained so far reveal excellent receptiveness and acceptance of the machine management software for specialized gyms. Through interviews and form collection, it was possible to identify that the majority of users expressed interest in frequently using the software, demonstrating its adaptability to the target audience of specialized gyms. Furthermore, the results indicate user agreement regarding ease of use, function integration, quick system learning, and information organization. These aspects are fundamental to ensure usability and user satisfaction. These positive results reinforce the importance of applying technology in the context of specialized gyms focused on personalized training. The implementation of the machine management software has the potential to optimize management processes, offering a more efficient environment adapted to the needs of the gyms. This will provide a differentiated experience for users, contributing to the improvement of operational efficiency in gyms and customer satisfaction.

**Keywords:** Software; Agile Methodology; Gym; Machine Management; Physical Activity.



## LISTA DE FIGURAS

Figura 1 – Menu Lateral .....	31
Figura 2 – Quadros de Horário .....	32
Figura 3 – Gerar Aulas .....	33
Figura 4 – Tela de Aulas .....	34
Figura 5 – Tela de Aulas com uma aula .....	35
Figura 6 – Tela de Aula .....	35
Figura 7 – Tela de Aulas – Visão do Aluno .....	36
Figura 8 – Diagrama de Funcionalidades .....	38
Figura 9 – Menu Lateral .....	41
Figura 10 – Fichas de Exercício .....	41
Figura 11 – Exercícios da Ficha .....	42
Figura 12 – Treinos .....	43
Figura 13 – Avaliações .....	44
Figura 14 – Tela Inicial .....	46
Figura 15 – Tela Meu Movimento .....	47
Figura 16 – Tela Resultados .....	48
Figura 17 – Tela Resultados de Medida .....	49
Figura 18 – Tela Resultado de Recordes .....	50
Figura 19 – Tela de Perfil .....	51
Figura 20 – Modelo estrutural de tela de login .....	58
Figura 21 – Modelo estrutural da tela de Menu.....	59
Figura 22 – Modelo estrutural da tela de Cadastro.....	60
Figura 23 – Modelo estrutural da tela de Gerenciamento .....	61
Figura 24 – Diagrama de Caso de Uso .....	63
Figura 25 – Modelo Lógico .....	81
Figura 26 – Este fluxo está agradável visualmente? Respostas .....	85
Figura 27 – Ocorreu algum erro durante a execução do fluxo? Respostas .....	86
Figura 28 – Este fluxo foi intuitivo? Respostas .....	87
Figura 29 – Resultados questão 01 .....	90
Figura 30 – Resultados questão 09 .....	92
Figura 31 – Diagrama de Casos de Uso .....	95
Figura 32 – Tela de Geração de Aulas .....	96
Figura 33 – Tela de Adição de Aluno .....	98
Figura 34 – Tela de Gerenciamento de Exercícios .....	99
Figura 35 – Adição de Exercício à Ficha .....	100
Figura 36 – Visualização dos Exercícios .....	101
Figura 37 – Área inferior da página de Visualização Geral de Exercícios – Listagem de Exercícios .....	103
Figura 38 – Área superior da página de Visualização Geral dos Exercícios – Detalhes do Treinamento.....	104
Figura 39 – Área superior de softwares utilizados no mercado .....	104

Figura 40 – Área superior-central da Página de Visualização do Exercício Escolhido	105
Figura 41 – Área inferior da Página de Visualização do Exercício Escolhido antes de iniciar treinamento .....	106
Figura 42 – Área inferior da Página de Visualização do Exercício Escolhido após iniciar treinamento .....	106

## LISTA DE TABELAS

Tabela 1 – Fluxos Realizados pelo Administrador .....	64
Tabela 2 – Fluxos Realizados pelo Treinador Master .....	68
Tabela 3 – Fluxos Realizados pelo Treinador .....	72
Tabela 4 – Fluxos Realizados pelo Aluno .....	79
Tabela 5 – Semelhanças e diferenças.....	107

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>14</b>
1.1	CONTEXTUALIZAÇÃO.....	14
1.2	PROBLEMÁTICA E JUSTIFICATIVA.....	15
1.3	OBJETIVOS .....	17
<b>1.3.1</b>	<b>Geral</b> .....	<b>17</b>
<b>1.3.2</b>	<b>Específicos</b> .....	<b>17</b>
1.3.2.1	<i>Exploratórios</i> .....	18
1.3.2.2	<i>Descritivos</i> .....	18
1.3.2.3	<i>Explicativos</i> .....	18
<b>2</b>	<b>REVISÃO TEÓRICA</b> .....	<b>19</b>
2.1	TRABALHOS RELACIONADOS.....	28
<b>2.1.1</b>	<b>Estudo De Caso 1: Sistema Desenvolvido Por Acadêmico</b> .....	<b>28</b>
2.1.1.1	<i>Aplicativo mobile para gerenciamento de academias de treinamento funcional</i> .....	28
2.1.1.2	<i>Aplicativo móvel para academia: estudo de tecnologias e desenvolvimento</i> .....	36
<b>2.1.2</b>	<b>Estudo 2: Mercado</b> .....	<b>44</b>
2.1.2.1	<i>Aplicativo de treinamento da Selfit</i> .....	44
<b>3</b>	<b>PROPOSTA E METODOLOGIA</b> .....	<b>52</b>
3.1	METODOLOGIA.....	52
<b>4</b>	<b>MODELAGEM E IMPLEMENTAÇÃO</b> .....	<b>54</b>
4.1	PROPÓSITO .....	54
4.2	ESCOPO .....	54
4.3	DEFINIÇÃO DO PRODUTO .....	55
<b>4.3.1</b>	<b>Perspectiva do produto</b> .....	<b>55</b>
<b>4.3.2</b>	<b>Funções do produto</b> .....	<b>55</b>
<b>4.3.3</b>	<b>Características do usuário</b> .....	<b>55</b>

<b>4.3.4</b>	<b>Limitações</b>	<b>56</b>
<b>4.3.5</b>	<b>Definições</b>	<b>57</b>
<b>4.4</b>	<b>IMPLEMENTAÇÃO</b>	<b>57</b>
<b>4.4.1</b>	<b>Interface Externa</b>	<b>57</b>
4.4.1.1	<i>Tela de Login</i>	57
4.4.1.2	<i>Tela de Menu</i>	58
4.4.1.3	<i>Tela de Cadastro</i>	59
4.4.1.4	<i>Tela de Gerenciamento</i>	60
<b>4.4.2</b>	<b>Funções</b>	<b>61</b>
4.4.2.1	<i>Requisitos do usuário</i>	61
4.4.2.1.1	<u>Requisitos funcionais</u>	61
4.4.2.1.2	<u>Regras de negócio</u>	62
4.4.2.1.3	<u>Requisitos não funcionais</u>	62
4.4.2.1.4	<u>Ferramentas utilizadas</u>	62
4.4.2.2	<i>Diagrama de caso de uso</i>	63
4.4.2.3	<i>Descrição dos casos de uso</i>	64
4.4.2.3.1	<u>Administrador</u>	64
4.4.2.3.2	<u>Treinador Master</u>	68
4.4.2.3.3	<u>Treinador</u>	72
4.4.2.3.4	<u>Aluno</u>	79
4.4.2.4	<i>Modelo Lógico</i>	81
<b>5</b>	<b>RESULTADOS</b>	<b>82</b>
5.1	TESTE BETA	82
5.1.1	<b>Questionário</b>	<b>82</b>
5.1.2	<b>Análise</b>	<b>83</b>
5.2	TESTE DE USABILIDADE E QUALIDADE DE SOFTWARE	87
5.2.1	<b>Questionário</b>	<b>88</b>
5.2.2	<b>Entrevistas</b>	<b>89</b>
5.2.3	<b>Análise</b>	<b>89</b>

5.3	SEMELHANÇAS E DIFERENÇAS .....	94
<b>5.3.1</b>	<b>Requisito de Bacharel .....</b>	<b>94</b>
5.3.1.1	<i>Aplicativo mobile para gerenciamento de academias de treinamento funcional</i> .....	94
5.3.1.2	<i>Aplicativo móvel para academia: estudo de tecnologias e desenvolvimento</i> .....	98
<b>5.3.2</b>	<b>Mercado .....</b>	<b>101</b>
5.3.2.1	<i>Aplicativo de treinamento da Selfit .....</i>	101
<b>5.3.3</b>	<b>Tabela de Semelhanças e Diferenças.....</b>	<b>106</b>
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>108</b>
6.1	MELHORIAS FUTURAS .....	109
6.1.1	<b>Treinador Master com acesso a fluxos de treinador .....</b>	<b>109</b>
6.1.2	<b>Plano de aulas .....</b>	<b>110</b>
6.1.3	<b>Permitir ao aluno realizar a edição de treinamento .....</b>	<b>110</b>
6.1.4	<b>Importação .....</b>	<b>111</b>
6.1.5	<b>Integração com outras plataformas .....</b>	<b>111</b>
6.1.6	<b>Permitir ao aluno realizar a troca de horário .....</b>	<b>112</b>
	<b>REFERÊNCIAS .....</b>	<b>113</b>

# 1 INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO

Em um mundo onde saúde e corpos perfeitos são metas de uma grande parte da população, independentemente da idade, a cultura esportiva modificou-se para suprir este objetivo. Como bem citado pela coordenadora do curso de Educação Física da Universidade de Fortaleza, professora Diane Nocrato (2020, n.p, grifo nosso), “*a promoção da saúde está em alta, o profissional teve que se adaptar e aprender a trabalhar a tecnologia ao seu favor e [a favor] do aluno*”. Assim, destacam-se no mercado de trabalho aqueles profissionais que incorporam novas tecnologias ao trabalho, facilitando o acesso da população à saúde corporal.

Nesse contexto, pessoais trainers, em sua atividade profissional, necessitam, segundo Silva e Oliveira (2017, p. 116), de “habilidade tátil, inteligência, treinamento acadêmico, experiência prática, julgamento correto e excelentes habilidades de comunicação”. Diante de tal cenário, ferramentas que facilitem a comunicação do profissional com o aluno tornam-se extremamente relevantes. Assim, segundo Ralph Vianna, gerente de Marketing e autor do blog da IESPE, Instituto de renome em Juiz de Fora,

Mobilidade sempre é um desafio, mas a tecnologia que aproxima o Profissional de Educação Física do aluno já está amplamente acessível [...] Os apps facilitam uma maior capacidade de organizar séries personalizadas, de gerenciar uma grande quantidade de alunos, de elaborar treinamentos, de calcular a rentabilidade etc (VIANNA, 2022, n.p).

A ideia inicial deste aplicativo surgiu, então, nesse contexto, buscando a implementação de tecnologias automatizadas – a exemplo da definição de horários e feedback dos atendimentos – e propondo soluções para otimizar o tempo do personal.

Porém, com o desenvolvimento do problema e contato com profissionais que trabalham integralmente na área, identificamos, a partir de entrevistas informais, uma problemática ainda maior do que a necessidade anterior, a saber: a falta de comunicação não verbal entre treinadores de uma mesma academia para alinhamento do uso de

recursos, gerando estresse no ambiente de trabalho e confusão em situações de alto fluxo de alunos. A identificação deste problema desaguou, então, num novo objetivo para o software.

A ideia do software passou a ser, então, promover uma comunicação visual, intuitiva e simples para identificação de aparelhos livres, assim como facilitar o gerenciamento de atendimentos. O público alvo principal são academias voltadas apenas para treinamento personalizado, sendo ativamente utilizado pelos pessoais trainers.

Para realizar o desenvolvimento do software da forma mais otimizada possível, foi decidido utilizar, seguindo os passos de Aléssio; Sabadin; Zanchett (2017, p. 3, grifo nosso), uma “*abordagem de desenvolvimento de software elaborada com disciplina e métodos bem definidos*”, mais conhecida como Engenharia de Software, utilizando técnicas como Análise de Requisitos, Design, Implementação e Testes. Cabe observar, assim, que, já que este trabalho não possui interesse de continuidade do projeto, não será realizada a etapa de Evolução.

Para o desenvolvimento do software, já na etapa de Implementação, selecionamos o modelo de Processo Ágil Scrum, método apresentado inicialmente no artigo “The New Product Development Gamepor” pelos autores Hirotaka Takeuchi e Ikujiro Nonaka. Neste modelo, são selecionadas as tarefas a serem desenvolvidas, seguindo os princípios do desenvolvimento ágil, em ciclos chamados Sprints, de modo que, após a finalização de cada Sprint, será analisada a qualidade das entregas e o cumprimento dos prazos, reajustando o planejamento do projeto quando necessário.

## 1.2 PROBLEMÁTICA E JUSTIFICATIVA

Atualmente o interesse na vida saudável e o culto a corpos perfeitos é extremamente comum, trazendo uma necessidade crescente de academias e pessoais trainers. Como se relata Walter Thompson, em seu artigo publicado no jornal do Colégio Americano de Medicina do Esporte, o treinamento personalizado está como uma das vinte tendências do universo fitness que devem estar em alta, levando em consideração



a opinião de diversos profissionais da saúde e da educação física. Neste contexto, academias especializadas no treinamento personalizado surgiram de forma emergente e não estruturada, donde surge a problemática a ser explorada neste trabalho: a desorganização no gerenciamento de recursos.

Como experienciado pela aluna e gestora do projeto, em horários com alto fluxo de alunos acompanhados por diferentes treinadores, é muito difícil identificar qual máquina está sendo utilizada, já que esta comunicação é feita de forma verbal, trazendo um ambiente com alta poluição sonora (música, conversas paralelas, conversas a respeito do uso de máquinas etc.).

A problemática que surge a partir da pergunta "Como desenvolver um software de gerenciamento de máquinas para academias especializadas em treinamento personalizado?" diz respeito aos desafios e obstáculos envolvidos no desenvolvimento de um software que atenda às necessidades específicas das academias especializadas em treinamento personalizado.

Ao abordar essa problemática, é importante considerar os seguintes aspectos:

**Requisitos específicos:** As academias especializadas em treinamento personalizado possuem características distintas em termos de suas máquinas, programas de treinamento, necessidades de controle e acompanhamento dos alunos. O desenvolvimento do software deve levar em conta esses requisitos específicos para garantir a eficácia e adequação do sistema.

**Controle de máquinas:** Um dos principais desafios é desenvolver um software capaz de gerenciar as máquinas presentes na academia, controlando seu uso, disponibilidade, manutenção e garantindo a segurança dos usuários. É necessário criar uma interface intuitiva e eficiente para facilitar o acesso e o controle dessas informações.

**Personalização e individualização:** As academias especializadas em treinamento personalizado têm como objetivo principal oferecer programas de treinamento adaptados às necessidades individuais de cada aluno. O software precisa ser capaz de fornecer recursos que permitam o registro e o acompanhamento personalizado de cada aluno, possibilitando a criação de treinos específicos e o monitoramento de progresso.

Integração de funcionalidades: Além do gerenciamento de máquinas e controle de alunos, é importante considerar a integração de outras funcionalidades que possam agregar valor ao software, como o controle de frequência, histórico de treinos, controle de medidas do aluno, avaliação do treinamento, entre outros.

Usabilidade e interface intuitiva: Para que o software seja adotado e utilizado de forma efetiva pelas academias, é fundamental garantir uma interface amigável, intuitiva e de fácil utilização. Os usuários devem conseguir navegar e realizar suas tarefas de forma ágil e sem complicações, mesmo sem um conhecimento técnico avançado.

Diante desses desafios, a problemática central é encontrar uma abordagem metodológica adequada e eficiente para o desenvolvimento e validação desse software de gerenciamento de máquinas, considerando as necessidades específicas das academias especializadas em treinamento personalizado, a integração de funcionalidades relevantes e a garantia de usabilidade e interface intuitiva.

Ao final deste projeto, entregaremos software testado e aprovado com a solução para a realização não verbal da comunicação a respeito das máquinas a serem utilizadas.

### 1.3 OBJETIVOS

O objetivo foi dividido em dois grandes grupos: o geral e os específicos, cabendo notar que estes últimos dizem respeito, em separado, ao projeto e ao produto. Em conjunto, tais ações específicas abarcam o objetivo geral do trabalho. A seguir, são identificados.

#### 1.3.1 Geral

Desenvolver uma aplicação web para academias especializadas em treinamento personalizado.

#### 1.3.2 Específicos

A presente pesquisa visa aos seguintes objetivos específicos:

#### *1.3.2.1 Exploratórios*

Pesquisar sobre/Aplicar hospedagem em um servidor web;

Pesquisar sobre as principais funcionalidades de um aplicativo para academias através de soluções semelhantes.

#### *1.3.2.2 Descritivos*

Descrever e explorar a tecnologia que será utilizada para desenvolvimento do software.

#### *1.3.2.3 Explicativos*

Desenvolver uma aplicação web em Python utilizando o Framework Django.

## 2 REVISÃO TEÓRICA

Para a implementação de um software que tenha como objetivo solucionar a problemática discutida anteriormente, seria necessário realizarmos abordagens para o alinhamento entre o produto esperado e o produto final, razão pela qual nos valem da Engenharia de Software, ou seja,

Um rebento da engenharia de sistemas e de hardware [...] abrange um conjunto de três elementos fundamentais - métodos, ferramentas e procedimentos - que possibilita [*sic*] ao gerente o controle do processo de desenvolvimento do software e oferece [*sic*] ao profissional uma base para a construção de software de alta qualidade produtivamente (PRESSMAN, 1995, p. 31).

Assim, baseando-se na fala de Sommerville (2011, p. 4) “Um bom software deve prover a funcionalidade e o desempenho requeridos pelo usuário; além disso, deve ser confiável e fácil de manter e usar”.

Segundo Bordin (2021, p. 4), os requisitos de usuário são “declarações em linguagem natural com diagramas dos serviços que o sistema deverá fornecer e suas restrições operacionais”, escrito para que os clientes possam compreender e validar se, a partir destas regras, a sua necessidade é suprida, sem muitos detalhes técnicos. A partir daí, fizemos a classificação deste requisito, diferenciados em: Funcionais; Não-funcionais; e Regras de negócio.

De acordo com Karl Wieggers e Joy Beatty, os requisitos funcionais são especificações de comportamento do sistema, descrevendo o que o sistema fará em resposta a determinadas entradas. Esses requisitos são declarações de serviços que o sistema deve fornecer, detalhando como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Além disso, em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer (WIEGERS; BEATTY, 2013, p. 9).

Estes requisitos podem variar de acordo com o tipo de usuário, o tipo de aplicação que está sendo desenvolvida e em que aparelho o software será mais utilizado. Por conta disso, é de grande importância que se conheça o cliente e que se entendam as suas necessidades para, só então, iniciar o desenvolvimento de tais requisitos, diminuindo as chances de retrabalho, cabendo notar ainda que esses requisitos podem ser expressos de diversas maneiras e em diferentes níveis de detalhamentos. De forma geral e de acordo com Martins (2007, p. 209) "Os requisitos funcionais são aqueles que definem o comportamento do sistema, capturados através de casos de uso, que documentam as entradas, os processos e as saídas geradas".

Teoricamente, a especificação de requisitos funcionais tem por objetivo definir todas as necessidades do usuário e refletir as decisões de especificação tomadas. Aliando teoria e prática, porém, podemos dizer que, não havendo definições contraditórias nem modificações às regras, é quase impossível atingir a consistência e a completude dos requisitos, de tal maneira que, segundo Sommerville (2011, p. 60), "Os problemas podem surgir após uma análise mais profunda ou depois de o sistema ter sido entregue ao cliente". Esta divergência à teoria acontece devido à complexidade de um software e porque as pessoas possuem diferentes pontos de vista em relação a um problema, então, após uma análise mais aprofundada e com a visualização do produto mais tangível, os requisitos podem ser modificados, adicionados ou excluídos. De acordo com Pressman e Maxim (2021, p. 241), é crucial reconhecer que as inconsistências nos requisitos podem surgir ao longo do processo de desenvolvimento de software, especialmente quando as partes interessadas têm diferentes perspectivas e compreensões do problema. Por isso, destaca-se a importância de uma análise contínua e revisão dos requisitos, a fim de identificar e resolver tais inconsistências antes que se tornem problemas mais graves durante a entrega do sistema ao cliente.

Os requisitos não funcionais abrangem uma perspectiva mais abrangente do sistema como um todo, e, de acordo com Chung et al (1999, p.7), eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área. Ao contrário dos requisitos funcionais, eles não visam descrever

funcionalidades ou partes específicas do sistema, mas sim descrever características que estarão presentes em todo o sistema, podendo ter implicações legislativas, empresariais, inovadoras, entre outras.

Segundo Martins (2007, p. 209),

Os requisitos não funcionais são compostos por características não necessariamente associadas ao comportamento, como: Usabilidade: facilidade de utilização pelas pessoas, aspectos visuais, documentação e material de treinamento. Confiabilidade: robustez, proteção contra falhas, recuperação em caso de erro e precisão. Performance: o sistema deve apresentar disponibilidade, tempo aceitável de execução de transações, bom tempo de resposta, baixo uso de memória e outros recursos computacionais. Suporte: estes requisitos estão associados à facilidade de se manter o sistema em produção, definindo plataforma, linguagem, ambiente de desenvolvimento e testes, dentre outros.

Vê-se, portanto, que eles possuem um caráter abrangente dentro do sistema e, por conta disso, assumem, na maioria das vezes, mais importância que os requisitos funcionais individuais.

Por fim, falaremos das regras de negócio. Resumindo as ideias de Bordin (2021, p. 7), estas guiam como as funcionalidades do sistema devem se portar e, por isso, estão, em sua maioria, associadas a requisitos funcionais, não havendo exclusividade em ambas as partes. Inicialmente, ao se criar os requisitos funcionais, estas regras, normalmente, trazem consequências – alterações – no fluxo imaginado, trazendo restrições, fluxos alternativos, associações com outras funcionalidades, entre outras características, complementando o sistema para agir de acordo com o esperado pelos futuros usuários.

Segundo Bordin (2021, p. 7), seria justamente por conta destas “políticas de funcionamento de operações no sistema e na organização, ou seja, regras respeitadas pelo sistema” que conseguimos “forçar” o usuário a utilizar a ferramenta de uma forma correta. Por conta disso, é de extrema importância que tais políticas não tenham ambiguidade, cabendo assinalar que, caso possuam funcionalidades agregadas para que estas exerçam as regras, haverá a necessidade de uma revisão das regras de negócio antes e ao final do desenvolvimento da funcionalidade agregada.

Finalizada a etapa inicial – tendo em vista a possibilidade de alteração no decorrer do projeto – da especificação dos requisitos, passamos para a etapa de análise e especificação de requisitos do sistema para, assim, segundo Martins (2007, p. 216), "traduzir os requisitos numa especificação que indica como o sistema deve ser implementado".

Realizamos a análise dos requisitos coletados para, de acordo com Zanchett (2015, p. 67), "identificar, quantificar, definir, especificar, documentar, rastrear, priorizar e classificar os principais problemas que o futuro software deve resolver", e também "criando um modelo conceitual do modelo de objetos" (MARTINS, 2007, p. 216). Esses documentos podem ser classificados em: 1) Requisitos Estruturais, que, segundo Larman (2004, p. 145), exibem a organização de um sistema em termos de seus componentes e relacionamentos, mostrando a estrutura do projeto do sistema e a organização do sistema quando em execução; ou 2) Requisitos Comportamentais, que são modelos do comportamento dinâmico do sistema quando em execução, mostrando o que acontece ou deve acontecer quando o sistema responde a um estímulo de seu ambiente.

Para realização dos modelos, selecionamos a Linguagem de Modelagem Unificada (UML), a qual, de acordo com Martins (2007, p. 158), "é uma linguagem padrão para documentar projetos de software. O UML pode ser utilizado para visualizar, especificar, construir e documentar os elementos de um sistema baseado em software". Zanchett (2015, p. 70), nesse aspecto, aponta que "muitas organizações utilizam a modelagem para contextualizar um projeto".

Guiando-nos para o desenvolvimento de toda documentação necessária para a implementação do software –o que nos habilita a seguir para a próxima etapa –, utilizamos a Modelagem do Diagrama de Casos de Uso e os Documentos gerados a partir dele para definir as Tarefas que serão tratadas nas Sprints, quando o desenvolvimento estiver na fase de aplicação de funcionalidades específicas.

Seguindo a proposta de Sommerville (2011, p. 86), utilizamos a "Modelagem de caso de uso, usada principalmente para modelar interações entre um sistema e atores externos (usuários ou outros sistemas)". Analisamos, então, os requisitos a fim de buscar

atores e funcionalidades pertencentes ao sistema, e em seguida representamos, utilizando a UML de forma clara e simples, as funcionalidades presentes no software e seus fluxos, assim como a interação das ferramentas com os atores (utilizadores funcionalidades do sistema).

Com o diagrama criado, partimos para o documento de caso de uso, em que se

especifica um comportamento externo de um sistema descrevendo um conjunto de sequências de ações realizadas pelo sistema para produzir um resultado de valor observável por um ator, realizado através de uma interação típica entre um ator (usuário, outro sistema computacional ou um dispositivo) e um sistema (Zanchett, 2015, p. 70).

Tal cenário serve como uma espécie de roteiro da funcionalidade específica do sistema.

A técnica de documentação dos Casos de Uso foi proposta por Ivar Jacobson em sua metodologia de desenvolvimento de sistemas orientados a objetos, podendo-se dizer que um caso de uso é, segundo Santos et al. (2006, p. 71): "[um] documento narrativo que descreve a seqüência [sic] de eventos de um ator (um agente externo) que usa um sistema para completar um processo". Seus principais objetivos são:

Serem compreensíveis para usuários que, provavelmente, não entendem de informática; Incentivarem a análise do sistema, especificando as funcionalidades necessárias; Delimitarem o sistema; Servirem de base para os casos de teste (Santos et al., 2006, p. 73).

Este documento mostra-se imprescindível para o desenvolvimento do software, pois, segundo o mesmo periódico (SANTOS et al., 2006, p. 73), "Os casos de uso permitem verificar se o desenvolvedor e o usuário concordam sobre o que o sistema deve fazer" facilitando a comunicação entre usuário e programador.

Finalizada a documentação dos Casos de Uso, iniciamos a etapa de estudo das classes necessárias para a implementação do software e cumprimento dos requisitos de usuário, criando um Modelo Conceitual Estrutural Orientado a Objetos, o qual figura como



um Diagrama de Classe, tendo em vista que o software será implementado seguindo o modelo de programação orientada a objeto. Importante lembrar, nesse contexto, que

Os diagramas de classe são usados no desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes de um sistema e as associações entre essas classes. Em poucas palavras, uma classe de objeto pode ser pensada como uma definição geral de um tipo de objeto do sistema. Uma associação é um link entre classes que indica algum relacionamento entre essas classes. Consequentemente, cada classe pode precisar de algum conhecimento sobre sua classe associada (SOMMERVILLE, 2011, p. 90).

Além disso, é possível refinar a análise e os modelos conceituais anteriores em diagramas de classe que mostrem as partes específicas do sistema, interfaces com o usuário, implementações lógicas e assim por diante. Os diagramas de classe tornam-se, então, uma captura instantânea que descreve exatamente como o sistema funciona, os relacionamentos entre os componentes do sistema em vários níveis e como planeja implementar esses componentes (IBM, 2021).

Por conta da facilidade agregada ao desenvolvedor quando criarmos o Modelo Conceitual Estrutural, analisaremos os casos de uso em busca de Classes e Métodos para, em seguida, estruturá-lo no Diagrama de Classes.

A junção do Diagrama de Classes com o Diagrama de Casos de Uso e os Protótipos de Tela, no ciclo de desenvolvimento, criam a base para as delimitação das Tarefas, que é, segundo Audy (2015, p. 44), uma “unidade de documentação que declara cada um dos requisitos da solução desejada, escritos pela perspectiva dos usuários ou stakeholder (envolvidos), indicando quem quer, o que e o porquê”, sendo, então, conforme Lerche-Jensen (2019, n.p.) “decomposto num nível que dê ao Time Scrum informações adequadas para criar entregáveis”. Assim, serão escolhidas inúmeras tarefas para serem realizadas no início do ciclo de desenvolvimento, as chamadas Sprints, criando

Esta lista detalhada contém todas as tarefas que o Time Scrum se comprometeu para a Sprint atual. Ela contém descrições sobre cada tarefa, juntamente com estimativas derivadas durante o processo de Criação de Tarefas. A Lista de Tarefas deve incluir quaisquer esforços de testes e integrações para que a

Incrementação do Produto seja integrada com sucesso em entregáveis das últimas Sprints (LERCHE-JENSEN, 2019, n.p.).

No time Scrum, vários papéis são necessários para que as diferentes perspectivas do desenvolvimento sejam obedecidas, tendo cada papel uma responsabilidade. Os papéis principais e suas missões são, segundo Sabbagh (2013, p. 39): 1) “Product Owner, [...] responsável por definir, comunicar e manter a Visão do Produto relativamente constante ao longo do projeto”, ocupado por apenas uma pessoa. O Product Owner, ainda de acordo com Sabbagh (2013, p. 39) “trabalha com os clientes do projeto e com quaisquer outras partes interessadas que possam contribuir para o entendimento e definição da Visão do Produto”; 2) o Time de Desenvolvimento, responsável por

Realizar o trabalho de desenvolvimento do produto, é multidisciplinar, o que significa que possui, em seus membros, todo o conhecimento necessário para realizar esse trabalho. O Time de Desenvolvimento é também auto-organizado, ou seja, ele próprio define como irá realizar o trabalho e gerenciar seu progresso em direção a metas de negócios acordadas com o Product Owner (SABBAGH, 2013, p. 40).

E, por fim, Scrum Master, responsável por garantir que os impedimentos que o Time de Desenvolvimento encontre em seu trabalho sejam removidos, atuando, quando necessário, como um agente de mudança na organização (SABBAGH, 2013, p. 40).

Segundo Sabbagh (2013, p. 41), “O Time de Scrum então está pronto para iniciar o primeiro de vários ciclos do projeto, nos quais o trabalho de desenvolvimento do produto será realizado. Esses ciclos são chamados de Sprints”. A Sprint

Representa um ciclo de trabalho e é um time-boxed de geralmente um mês, onde algum valor é acrescentado a um produto em desenvolvimento. Ou seja, ao final dela, o produto precisa ter algum incremento que gere valor aos olhos do cliente e que satisfaça a meta (COUTINHO, 2018, n.p.)

E, para que estes incrementos possam ser desenvolvidos corretamente, várias reuniões são necessárias para alinhamento dos papéis explicitados anteriormente, diminuindo retrabalho, impedimentos e alinhando as expectativas e necessidades de cada papel. As principais reuniões são: Planejamento da Sprint, Dailies, Inspeção e Revisão da Sprint.

O planejamento da Sprint ocorre de acordo com seu tamanho, ou seja, depende da necessidade de validação e complexidade do projeto, sendo iniciada a partir da reunião citada anteriormente, na qual:

Se planeja o trabalho a ser realizado no próprio Sprint. Nessa reunião, Time de Desenvolvimento e Product Owner negociam, a partir dos itens (também chamados de tarefas) [...]o que será desenvolvido. Ou seja, facilitados pelo ScrumMaster, eles selecionam um conjunto de itens [...] que julgam ser capazes de desenvolver na duração do Sprint, o que é apenas uma previsão, e estabelecem um objetivo ou meta de negócios a ser alcançada com o desenvolvimento desses itens, chamada de Meta do Sprint. O Time de Desenvolvimento, então, se compromete com atingir essa Meta do Sprint (SABBAGH, 2013, p. 42).

As Dailies acontecem, segundo Sabbagh (2013, p. 44), “Em cada dia do trabalho de desenvolvimento, os membros do Time de Desenvolvimento se encontram por no máximo quinze minutos, preferencialmente no mesmo horário e no mesmo local”. Apontamos ainda que “O objetivo da reunião é garantir a visibilidade de seu trabalho entre eles e planejar, informalmente, o próximo dia de trabalho. O Scrum Master pode, se necessário, facilitar essa reunião, mas sua presença não é obrigatória” (SABBAGH, 2013, p. 44-45).

Para dinamizar a reunião e cumprir o limite de tempo, estruturamos as falas do Time de Desenvolvimento em três tópicos principais: o que foi feito desde a última daily; o que será feito até a próxima daily; e o que impede a conclusão das tarefas desta daily —, realizando, assim, o alinhamento da equipe e unindo impedimentos a soluções, o que é importante para o progresso rápido do projeto.

Assim como a reunião de planejamento da Sprint, o contexto de encerramento da Sprint está diretamente ligado ao seu contexto, porém, este encerramento conta com duas reuniões, marcando o encerramento do ciclo: Revisão da Sprint e Retrospectiva da Sprint. Na reunião de Revisão da Sprint, segundo Sabbagh, (2013, p. 46), “faz-se a inspeção e adaptação do produto”. Já na Retrospectiva da Sprint, Sabbagh (2013, p. 46) afirma que: “faz-se a inspeção e adaptação da forma de trabalhar do Time de Scrum”.

Na reunião de Revisão da Sprint

O Product Owner convida ... os clientes do projeto e demais pessoas relevantes que possam prover feedback sobre o que foi produzido durante o Sprint. Nessa reunião, o Time de Desenvolvimento e o Product Owner apresentam e demonstram para os presentes o que foi produzido no Sprint, para então deles obter feedback (SABBAGH, 2013, p. 46).

Após coletados feedbacks, então, é realizado

Um acordo entre Product Owner e Time de Desenvolvimento sobre o que é necessário para que um item ou o Incremento do Produto como um todo seja considerado pronto e, assim, passe a fazer parte do produto em desenvolvimento (SABBAGH, 2013, p. 46).

Apresentamos na reunião de Revisão da Sprint estes itens que foram considerados prontos.

Após a reunião de Sprint Review, Time de Desenvolvimento e Product Owner realizam a reunião de Sprint Retrospective, facilitados pelo ScrumMaster. Nela, ambos identificam o que foi bem no Sprint corrente, e que por essa razão pode ser mantido no próximo Sprint, e o que pode melhorar, buscando formas práticas e traçando planos de ação para fazê-lo. O objetivo desse processo de melhoria contínua é tornar o Time de Scrum cada vez mais efetivo. Uma vez terminada a reunião de Sprint Retrospective, está encerrado o Sprint. Um novo Sprint se inicia imediatamente após o término da anterior (SABBAGH, 2013, p. 47).

Este processo em busca de automelhoria constrói uma equipe que, ao final do projeto, visualiza mais claramente suas qualidades e defeitos, assim como formas mais ágeis de resolver problemas.

Após o desenvolvimento, é iniciada a fase de testes beta, na qual um terceiro realiza os passos especificados no documento de casos de uso, analisando o fluxo em busca de erros, possíveis melhorias e avaliando a interface do sistema. Essa etapa é de extrema importância antes de lançar o software para o público, pois, como afirma Nielsen (1993), a qualidade do software desempenha um papel crucial no desenvolvimento de um sistema. Problemas de usabilidade identificados pelos usuários podem prejudicar seriamente a reputação como fornecedor de produtos de qualidade. Além disso, as solicitações de alteração decorrentes de problemas de usabilidade apontados pelos

usuários costumam ser muito mais dispendiosas do que as alterações feitas durante as fases iniciais do projeto. Portanto, torna-se essencial investir na qualidade do software desde o início, a fim de evitar problemas futuros e de garantir a satisfação dos clientes.

## 2.1 TRABALHOS RELACIONADOS

Esta seção tem como objetivo apresentar algumas aplicações de software desenvolvidas anteriormente, tanto em meio acadêmico quanto no mercado, que possam contribuir de alguma forma para este trabalho. Essas aplicações foram selecionadas com base na estrutura buscada, visando gerenciar os atendimentos dos treinadores e os exercícios a serem realizados pelos alunos.

### 2.1.1 Estudo de Caso 1: Sistema Desenvolvido por Acadêmico

#### 2.1.1.1 *Aplicativo mobile para gerenciamento de academias de treinamento funcional*

Silva (2021) desenvolveu um aplicativo mobile para gerenciamento de academias de treinamento funcional voltado para Gestores de Academias Funcionais. A seguir, transcrevemos parte do resumo do seu trabalho:

Com uma solução que forneça acessibilidade através de um aplicativo mobile, a um custo acessível e que ajude os gestores de academias funcionais nas atividades mais básicas do cotidiano, o presente trabalho soluciona um problema ainda bastante prevalente no mercado. Ao final deste trabalho foi aplicado um questionário para o público alvo, e pode se perceber uma grande aceitação do aplicativo entre os entrevistados com 85.7% dando uma nota 7 ou maior quando perguntado se o aplicativo desenvolvido cumpre as funcionalidades necessárias numa solução como esta. (SILVA, 2021, n.p).

O objetivo do trabalho supracitado foi criar um aplicativo mobile que automatiza processos diários de academias que realizam treinamento funcional, tais como: cadastramento de quadro de horários; geração de aulas; disponibilização de aulas;

adição de alunos e confirmação de presença nas aulas. O aplicativo foi disponibilizado para a plataforma Android e IOS.

Para a realização deste objetivo, foram implementadas regras de negócio e requisitos funcionais e não funcionais que auxiliam na construção da aplicação.

Os requisitos funcionais, conforme afirma Sommerville, definem os possíveis usuários e quais as funcionalidades que esses usuários poderão utilizar, informações importantes para a construção e entendimento do produto. Abaixo, apresentamos os principais requisitos funcionais da aplicação construída.

- RF02 – O aplicativo deve permitir que o professor cadastre os horários de aulas (RN07).
- RF03 – O aplicativo deve permitir que o professor gere aulas a partir de dia e horário especificado (RN08).
- RF04 – O aplicativo deve permitir que o professor visualize as aulas (RN09).
- RF05 – O aplicativo deve permitir que o professor remova as aulas geradas (RN09). [...]
- RF07 – O aplicativo deve permitir que o aluno visualize as aulas geradas (RN10).
- RF08 – O aplicativo deve permitir que o aluno confirme sua presença em uma aula (RN02, RN11). [...]
- RF11 – O aplicativo deve permitir que o professor libere ou bloqueie as aulas geradas em um dia escolhido (RN14).
- RF12 - O aplicativo deve permitir que o professor disponibilize os exercícios que vão compor o treino (aula) do dia para o aluno antes da aula começar (RN15, RN16). [...]
- RF15 - O aplicativo deve permitir que professores disponibilizam dicas de saúde (RN20). [...]
- RF19 - O aplicativo deve permitir que professores e alunos possam visualizar as estatísticas de frequência e de progresso (RN25).
- RF20 - O aplicativo deve permitir que o gestor tenha controle financeiro sobre o pagamento dos alunos. (SILVA, 2021, p. 45).

Conforme análise dos requisitos funcionais, podemos identificar algumas características do produto criado, como o gerenciamento de usuários e a inserção de permissões vinculadas a estes para a visualização de funcionalidades pertinentes. As permissões identificadas pelos requisitos especificados acima são: Professor, Aluno e Gestor. Outro ponto identificado foi o usuário principal do software: o professor, já que a maioria dos requisitos funcionais o envolvem.

As regras de negócio, segundo Kamada (2006), podem ser definidas como declarações que definem ou restringem um aspecto do negócio. Elas objetivam a

estruturação do negócio e as influências comportamentais da aplicação, guiando as criações dos requisitos funcionais. Seguem as principais regras de negócio da aplicação construída.

RN02- Apenas professores podem cadastrar alunos. [...]

RN07- Ao cadastrar quadros de horários o professor define quais serão os horários de aulas que serão geradas (RN08) para os dias selecionados. Um quadro de horário pode ser definido como um conjunto de horários de aulas em um dia. Assim pode-se gerar aulas a partir deste quadro para vários dias, sem precisar gerar manualmente as aulas em diferentes dias repetidamente.

RN08- O aplicativo deve permitir que o professor gere aulas num período arbitrário de dias a partir de um quadro de horário (RN07) ou gere uma única aula para um único dia.

RN09- O aplicativo deve permitir que o professor visualize as aulas geradas, exclua e adicione novas aulas em um dia determinado.

RN10- O aplicativo deve permitir que o aluno veja as aulas geradas e disponibilizadas em qualquer dia vendo quantas vagas estão disponíveis naquela aula e quais usuários estão com presença confirmada e quem será o professor.

RN11- O aluno não poderá confirmar sua presença em uma aula cujo horário de fim já passou. [...]

RN14- O aplicativo deve permitir que o professor libere ou tranque as aulas geradas em um dia, disponibilizando-as ou não para os alunos. Nesta funcionalidade a cor dos quadros é pink quando bloqueadas e azul quando disponibilizada.

RN15- O aplicativo deve permitir que o professor insira os exercícios que serão praticados pelo aluno no treino do dia (na academia).

RN16- O aplicativo deve permitir que o professor insira exercícios para os alunos praticarem em suas casas, através de textos ou vídeos. [...]

RN20- O aplicativo deve permitir que o professor insira e disponibilize aos alunos textos e vídeos com dicas de saúde. [...]

RN25- O aplicativo deve permitir que professores e alunos informem suas medidas corporais, métricas de tempo e força nos exercícios e visualizem alteração dos mesmos ao longo do tempo (SILVA, 2021, p. 47).

Através das regras de negócio, entendem-se melhor as interações entre o usuário e a aplicação, como: os passos necessários para inclusão de aluno a uma ou mais aulas; a utilização do software para gerenciamento de aulas; quadros de horários por meio do professor; e a personalização dos usuários por meio da aplicação.

Após a análise dos objetivos do software e das especificações de como o software criado deve se portar, serão apresentadas telas da aplicação para visualização da realização destes objetivos (conferir Figura 1).

Figura 1 – Menu Lateral



Fonte: Silva (2021, p. 69)

Através do Menu, podemos identificar o sistema de permissionamento funcionando, por meio do qual o usuário da permissão Professor tem acesso aos: Quadros de Horário, Gerar Aulas, Aulas, Usuários (conferir Figura 2).

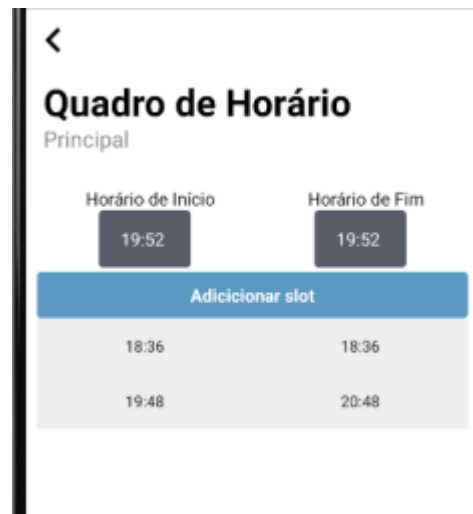


Figura 2 – Quadros de Horário



(a) Visualização Geral

(b) Adicionar novo



(c) Quadro de Horário

Fonte: Silva (2021, p. 70-71)

Através do Quadros de Horário, podemos visualizar o fluxo para o gerenciamento desta variável, sendo possível adicionar novos horários e visualizar os já criados. Na adição, podemos perceber que o único dado é o nome do quadro, já que os outros dados necessários para a geração de aulas são incluídos após sua criação, representada pela figura (c). Ao selecionar um quadro de horário já existente, podemos incluir os slots de tempo. Observemos ainda que, após esta configuração inicial, podem-se adicionar aulas, o que será apresentado a seguir (conferir Figura 3):

Figura 3 – Gerar Aulas

**Gerar aulas**

Principal

Período

01/09/21 04/09/21

Título

Metade da semana pra frente!

Descrição

Não esqueçam de trazer toalha!

Vacancies

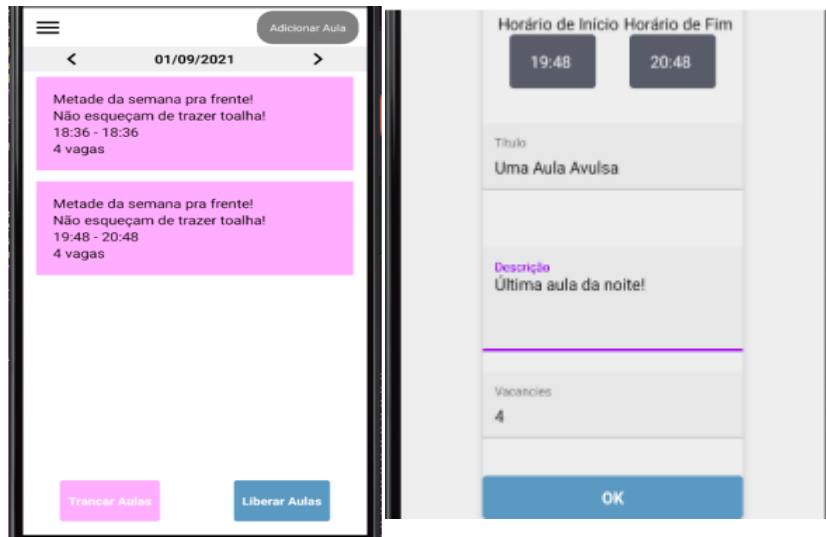
4

GERAR

Fonte: Silva (2021, p. 71)

Através do Gerar Aulas, podemos entender melhor como funciona a interação entre o quadro de horário e a geração de aulas: faz-se a seleção do primeiro campo e, quando geradas as aulas, estas possuirão os slots de tempo configurados através do quadro de horário. Outro campo importante para a configuração das aulas é o período, em que é delimitada a data de início e fim para a aula, sendo importante para a geração das aulas diárias. Além disso, destacamos o campo *vacancies*, que delimita o número máximo de alunos que podem se inscrever nas aulas diárias (conferir Figura 4).

Figura 4 – Tela de Aulas



(a) Visualização de aulas

(b) Adicionar Aula Avulsa

Fonte: Silva (2021, p. 72-73).

Através da tela de aulas (a), é possível identificar as aulas diárias, que foram geradas através da Figura 3, sendo necessária a liberação das aulas através do botão presente na página para que os alunos possam se inscrever. O trancamento só é possível se a aula estiver liberada, removendo os alunos já inscritos e impossibilitando a visualização e inscrição desta para os alunos.

Já na tela de adicionar aula avulsa, é necessário visualizar a data (a) em que se quer adicionar uma aula avulsa e selecionar o botão adicionar aula. Será criado, então, um novo slot de tempo desassociado do quadro de horários para a data selecionada, também podendo ser configurado o número de vagas (conferir Figura 5).

Figura 5 – Tela de Aulas com uma aula



Fonte: Silva (2021, p. 73)

Através da Tela de Aulas com uma aula avulsa e aulas liberadas, podemos visualizar o resultado final das configurações de aula, que ficarão na mesma tela independentemente de ser avulsa ou não (conferir Figura 6).

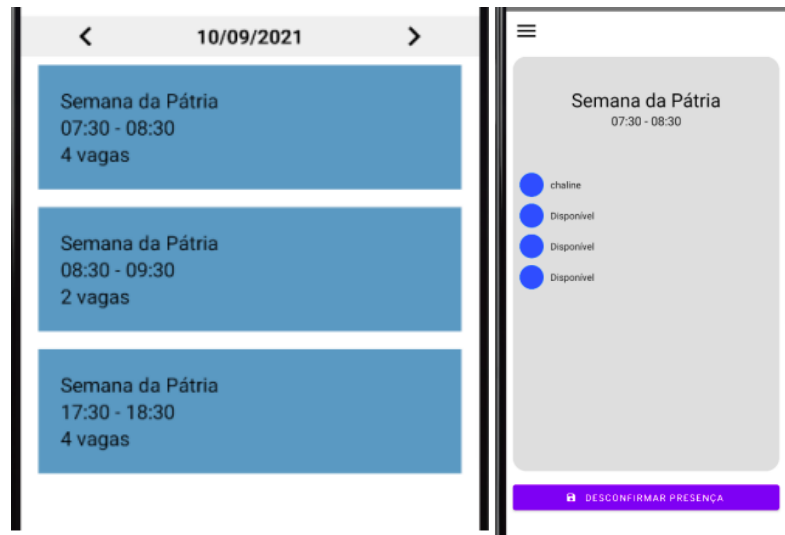
Figura 6 – Tela de Aula



Fonte: Silva (2021, p. 74)

Através da Tela de Aula, selecionada por meio da Figura 5, é possível visualizar os alunos que se inscreveram, e, caso haja vagas disponíveis, em vez do nome do aluno inscrito, apresenta-se novamente a palavra Disponível (conferir Figura 7).

Figura 7 - Tela de Aulas – Visão do Aluno



(a) Aulas disponíveis

(b) Alunos inscritos

Fonte: Silva (2021, p. 75-76).

Através da Tela de Aulas pela visão do aluno, podemos identificar uma diminuição nos fluxos possíveis para este tipo de usuário, cabendo notar que a tela será apenas utilizada para seleção da aula e confirmação ou desconfirmação da presença.

#### 2.1.1.2 Aplicativo móvel para academia: estudo de tecnologias e desenvolvimento

Soares (2016) desenvolveu um Aplicativo Móvel para Academia voltado para realizar estudo sobre as últimas tecnologias na área de aplicativos móveis, de modo a desenvolver um aplicativo contendo funções de gerenciamento de fichas de exercício, treinos e avaliações físicas. A seguir, é citado o resumo do seu trabalho:

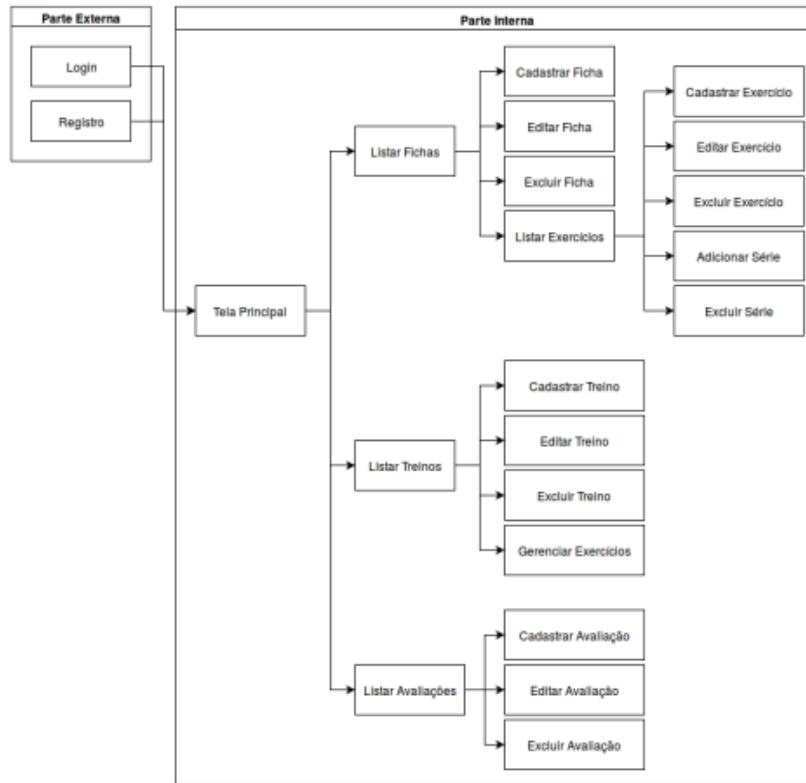
As tecnologias de informação estão se tornando cada vez mais essenciais na vida das pessoas, e com a popularização dos dispositivos móveis nos últimos

anos, há uma demanda por aplicativos que facilitam as tarefas do dia a dia de quem utiliza essa tecnologia. Nas academias, há a necessidade de fazer o gerenciamento das fichas de exercício dos seus alunos e um acompanhamento físico através de avaliações físicas. Entretanto, na maioria das vezes, todo esse gerenciamento é feito manualmente, o que gera um trabalho braçal enorme, se tornando uma tarefa cansativa para os professores, além de gerar gastos com papel. O objetivo desse trabalho foi fazer um estudo sobre as últimas tecnologias na área de aplicativos móveis, e com isso desenvolver um aplicativo contendo funções de gerenciamento de fichas de exercício, treinos e avaliações físicas. No final deste trabalho, foi possível obter um produto funcional e viável, contendo funcionalidades para auxiliar nas atividades da academia, além de conter características diferenciais como funcionamento offline e disponibilidade de dados (SOARES, 2016, n.p).

O objetivo deste trabalho foi realizar o estudo sobre a metodologia de criação de aplicativos móveis e, ao final, desenvolver um aplicativo voltado para o ambiente das academias. Ao realizar o desenvolvimento da aplicação, serão automatizados os seguintes processos: montagem de ficha de exercícios; e execução dos treinos nas academias.

Para o cumprimento do objetivo específico de desenvolvimento da aplicação, o autor lançou mão de um diagrama de funcionalidades do aplicativo, tornado possível visualizar o projeto como um todo e acompanhar o progresso do desenvolvimento (conferir Figura 8).

Figura 8 – Diagrama de Funcionalidades



Fonte: Soares (2016, p. 29)

Através da figura acima, podemos identificar que a estrutura do aplicativo é de fácil implementação, havendo apenas três fluxos principais que, ao se ramificarem, trazem, na maioria dos casos, as interações CRUD das tabelas criadas no Banco de Dados do software.

Além do diagrama citado anteriormente, foi realizado um documento de especificação de requisitos, em que estão documentadas todas as informações necessárias para a construção da aplicação, sendo tal documento inteiramente voltado aos requisitos funcionais e requisitos não funcionais. Como o documento traz com bastante detalhamento cada um dos requisitos e como alguns são comuns em aplicações Web, serão citados apenas os que possuem mais importância para o entendimento da aplicação.

A ficha de exercício é um “artefato que auxilia o aluno a se organizar durante a execução dos exercícios” (SOARES, 2016, p. 4). A partir dela, será apresentada a listagem de exercícios com suas especificações. Também podendo apresentar observações para que o aluno se lembre de alguns detalhes, este artefato serve como um instrumento de consulta. Seguem, abaixo, os requisitos funcionais relacionados à ficha de exercícios:

RF002-1 - Listar fichas: O aluno pode listar todas as suas fichas de exercícios cadastradas. A listagem apresenta os dados básicos de cada ficha. As fichas são ordenadas por Nome. Elas também disponibilizam opções de editar e excluir.

RF002-2 - Cadastrar fichas: O aluno pode inserir uma nova ficha de exercícios no sistema. São disponibilizados os campos Nome, e Dias da Semana. O campo Nome representa um identificador para a ficha. O campo Dias da Semana é um campo que disponibiliza os dias da semana e o aluno pode escolher vários valores.

RF002-3 - Editar fichas: O aluno pode alterar as informações de uma ficha previamente cadastrada. Deve ser disponibilizado um formulário semelhante ao de cadastro, incluindo opções para gerenciamento dos exercícios.

RF002-3a - Cadastrar exercícios: O aluno pode inserir um novo exercício na ficha. Deve ser disponibilizado um campo Nome, contendo o nome do exercício, e um campo Grupo Muscular em que o usuário pode escolher uma opção única de grupo muscular.

RF002-3b - Editar exercícios: O aluno pode alterar as informações de um exercício previamente cadastrado. Ele também pode adicionar ou remover séries no exercício. As séries podem ser de dois tipos: tempo ou peso. As séries de tempo disponibilizam um campo onde o usuário deve informar o tempo, em segundos, daquela série. As séries de peso devem disponibilizar dois campos, um para o número de repetições, e outro para o peso, em kg.

RF002-3c - Excluir exercícios: O aluno pode excluir um exercício da ficha.

RF002-4 - Excluir fichas: O aluno pode excluir uma ficha de exercício cadastrada (SOARES, 2016, p. 5).

O treino é um artefato bastante vinculado à ficha de exercícios, tendo em vista que “funcionam como uma espécie de histórico que mostra qual o dia e o horário [em] que o aluno realizou um treino e quais exercícios foram feitos naquele treino”. Abaixo, os requisitos funcionais relacionados aos treinos:

RF003-1 - Listar treinos: O aluno pode listar todos os seus treinos cadastrados. A listagem apresenta dados básicos do treino, tais como tempo total, número de exercícios realizados, e porcentagem de exercícios concluídos. Os registros são ordenados por data.

RF003-2 - Cadastrar treinos: O aluno pode inserir um novo treino no sistema. Deve ser disponibilizado um campo para selecionar a Data e um campo para selecionar o horário de início.



RF003-3 - Editar treinos: O aluno pode alterar as informações do treino previamente cadastrado. Deve ser disponibilizado um formulário semelhante ao de cadastro, incluindo campo para horário de término, e deve disponibilizar a listagem dos exercícios a serem realizados. O aluno pode marcar quais exercícios já foram finalizados.

RF003-4 - Excluir treinos: O aluno pode excluir um treino cadastrado (SOARES, 2016, p. 7).

A avaliação física é um módulo voltado ao acompanhamento corpóreo do aluno, e, a partir dele, é possível acompanhar os resultados físicos dos treinos no aluno. Seguem os requisitos funcionais relacionados às avaliações físicas:

RF004-1 - Listar avaliações físicas: O aluno pode listar todas as avaliações físicas cadastradas. A listagem irá apresentar alguns dados básicos e será ordenada por data.

RF004-2 - Cadastrar avaliações físicas: O aluno pode inserir uma nova avaliação no sistema. Deverão ser disponibilizados os campos Altura (em cm) e Peso (em kg). Além disso, devem ser disponibilizados campos para Dobras Cutâneas e Medidas Lineares (circunferências). Os campos de medidas transversais não serão incluídos. As medidas devem seguir o mesmo padrão do software Corpore Teste (Corpore Teste, 2016). (...)

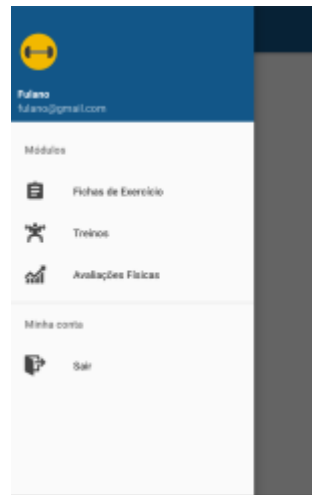
RF004-3 - Editar avaliações físicas: O aluno pode alterar as informações de uma avaliação previamente cadastrada. Deve ser disponibilizado um formulário semelhante ao do cadastro.

RF004-4 - Excluir avaliações físicas: O aluno pode excluir uma avaliação física cadastrada (SOARES, 2016, p. 8).

Conforme análise dos requisitos funcionais apresentados, podemos identificar algumas características do sistema, como a não necessidade de um sistema de permissão, já que será utilizado apenas pelo aluno. Outro ponto identificado no sistema é que os exercícios realizados nos treinos são passados através da ficha, relacionada com o treino a partir do dia da semana em que o treino foi aberto e da configuração do dia da ficha.

Após a análise das especificações de como o software criado deve se portar, serão apresentadas telas da aplicação para visualização da realização destes requisitos (conferir Figura 9).

Figura 9 – Menu Lateral



Fonte: Soares (2016, p. 18)

Através do menu lateral, podemos visualizar o ponto inicial de todos os fluxos presentes no aplicativo, percebendo-se, assim, a não necessidade de mais de um tipo de usuário para a aplicação. Também podemos analisar a utilização de ícones que facilitem ao usuário a memorização dos fluxos, facilitando a aprendizagem (conferir Figura 10).

Figura 10 – Fichas de Exercício



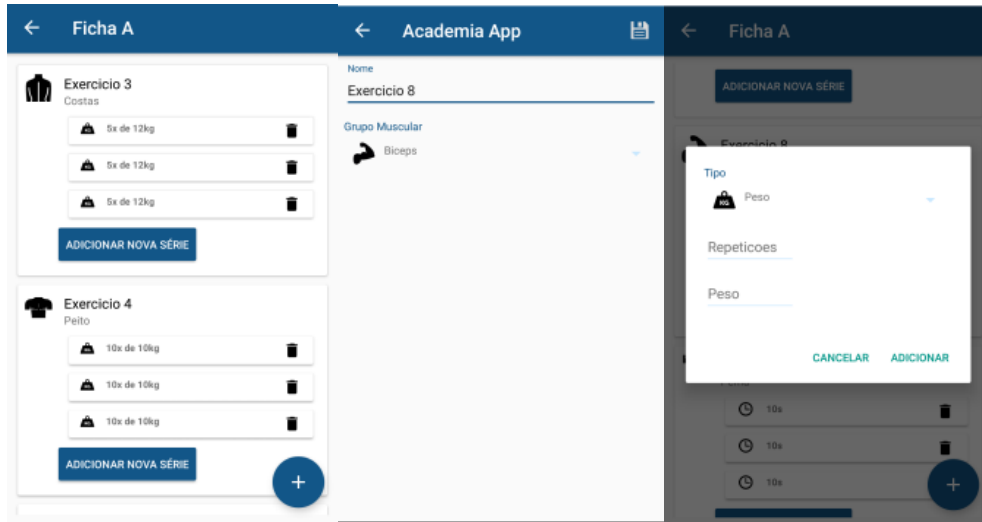
(a) Listar Fichas

(b) Criar nova Ficha

Fonte: Soares (2016, p. 19)

Através das Fichas de Exercício, é possível criar uma nova Ficha, configurando os dias em que se deve repetir os exercícios, inserindo um nome para identificação. Também é possível editar as fichas já cadastradas, adicionando os exercícios da ficha escolhida, fluxo visualizado a seguir (conferir Figura 11).

Figura 11 – Exercícios da Ficha

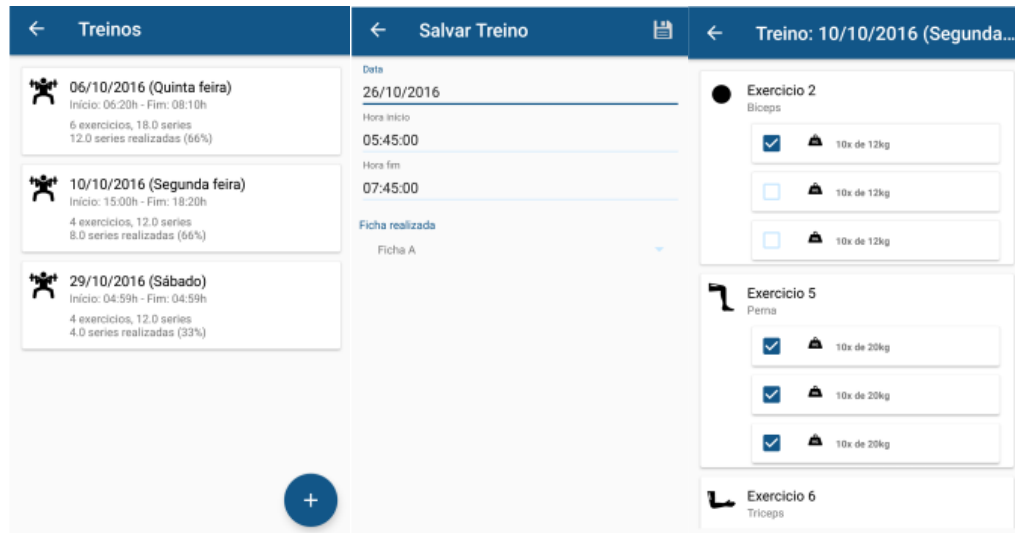


(a) Listar Exercícios (b) Adicionar novo Exercício (c) Adicionar Nova Série

Fonte: Soares (2016, p. 20)

Através dos Exercícios da Ficha, é possível visualizar os exercícios adicionados à ficha selecionada; adicionar novos, inserindo o nome do exercício e o grupo muscular envolvido; adicionar séries, inserindo o número de repetições e quantos quilos serão utilizados para a realização do exercício ou, caso seja um exercício do tipo chamado tradicionalmente de cárdio, pode-se informar o tempo de realização da atividade (conferir Figura 12).

Figura 12 – Treinos



(a) Listar Treinos (b) Adicionar novo Treino (c) Visualizar Exercícios do Treino

Fonte: Soares (2016, p. 21)

Através dos treinos, podemos visualizar os treinos salvos e adicionar novos, inserindo a data e o período de realização. Ao selecionar um treino salvo, é possível visualizar os exercícios a serem realizados neste treino, com base no dia da semana em que a data se encontra, e marcar a série que foi concluída (conferir Figura 13).

Figura 13 – Avaliações



(a) Listar Avaliações (b) Adicionar nova Avaliação

Fonte: Soares (2016, p. 22)

Através das avaliações, é possível visualizar aquelas realizadas e adicionar novas, preenchendo o formulário com as informações corpóreas do aluno, sendo também permitido visualizar as informações das avaliações anteriores – selecionando uma avaliação da lista, retornarão os dados já cadastrados.

## 2.1.2 Estudo de Caso 2: Mercado

### 2.1.2.1 Aplicativo de Treinamento da Selfit

A Selfit é uma empresa de academias em que há diversas aulas específicas, como dança, e o espaço para treinamento livre. Não há, portanto, treinamento personalizado: os treinadores enviam pelo aplicativo diversas listas de exercícios, cada uma delas tendo o tempo médio de execução de uma hora, e o aluno realiza, no espaço interno da academia, uma dessas listas. Há acompanhamento de medidas e peso dentro da

academia, além da marcação de horário para treinamento, acompanhamento dos exercícios e avaliação do treinamento. A atualização dessas informações, porém, é feita pelo próprio aluno, dando mais autonomia para que este selecione seus horários e atualize sua lista de exercícios. Transcrevemos, abaixo, a proposta da empresa exibida em seu site.

Acreditamos que treinar é mais que se exercitar, é um momento para liberar as energias, cuidar da saúde, estar leve para encarar a rotina de frente e se divertir. Desde 2012, trabalhamos com a missão de proporcionar uma melhor experiência para noss@s alun@s, dando suporte e liberdade para que tod@s possam expressar sua individualidade na hora do treino.

Seja o seu objetivo, perder peso, ganhar massa ou simplesmente se manter saudável, queremos que você se sinta bem e que faça tudo no seu ritmo. Aqui na Selfit, nós sabemos que cada um é cada um e que o exercício é pra todo mundo.

Somos a rede de academias que mais cresce no Brasil e estamos em constante evolução para oferecer a melhor estrutura, os equipamentos mais modernos e professores certificados para que cada alun@ encontre sua melhor versão todos os dias.

Mais que uma academia, somos amigos, parceiros e acolhedores.

Encontre a unidade mais próxima e #VemTreinar com a gente.

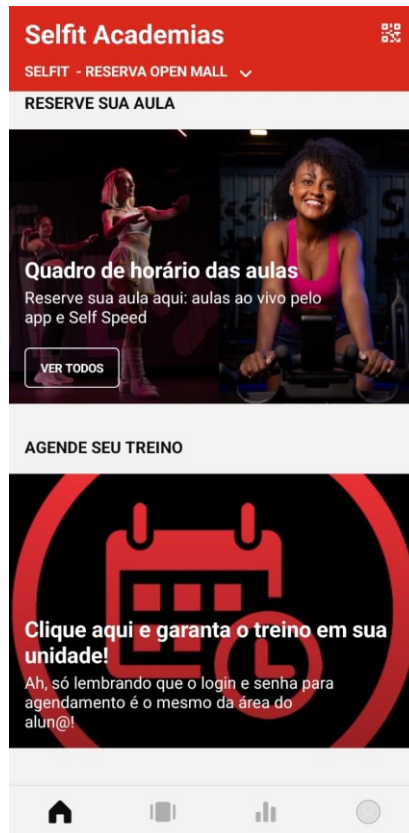
Seja bem-vind@ à academia do seu jeito. <3 (SELFIT, s.d., n.p).

A aplicação contém uma estrutura simples, voltada ao uso em dispositivos móveis e de fácil aprendizagem e memorização. Considerando o alto fluxo de alunos e novos entrantes, é imprescindível que o treinamento para utilização do software seja rápido ou nulo e, já que os alunos utilizam celulares smartphones ou os totens de impressão para visualizar o treinamento, não há necessidade de disponibilizar o software para desktops.

Por conta de o aplicativo ser comercial, as informações referentes à estruturação do projeto, como os requisitos ou casos de uso, não foram disponibilizadas para análise. Além deste, não é possível analisar a aplicação por meio de permissões que não sejam do Aluno, tendo em vista a necessidade do vínculo de funcionário à empresa. Logo, a aplicação será analisada apenas a partir da permissão do aluno.

A partir da estrutura da tela (conferir Figura 14), podemos visualizar a ferramenta que garante facilidade de acesso a todos os fluxos permitidos pelo usuário, ou seja, a barra horizontal inferior comumente chamada de barra de ferramentas.

Figura 14 – Tela Inicial

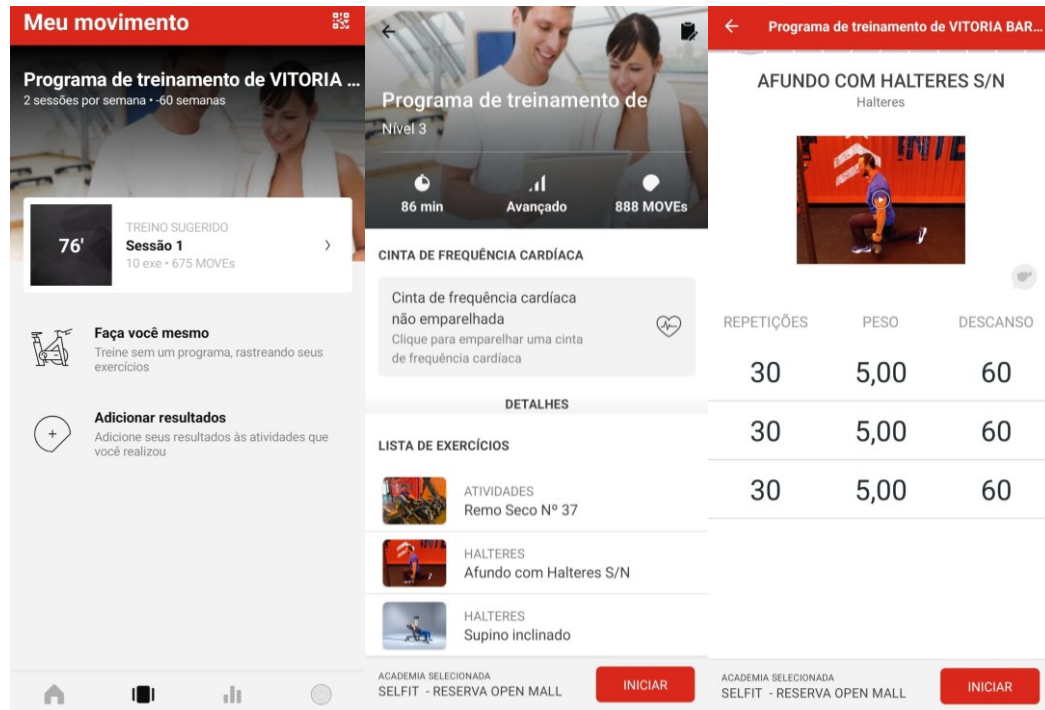


Fonte: Selffit (s.d., n.p)

Focando na tela inicial, é possível perceber que esta área é mais voltada para as informações e fluxos da academia que possam interessar ao aluno. Na área superior da tela, apresenta-se a filial selecionada para realizar treinos, a qual pode ser alterada através do clique na seta ao lado do nome da filial. Através desta página, podemos iniciar os fluxos de agendamento, seja de aulas (realizadas em grupo, com um profissional de Educação Física) seja de treinos (realizados individualmente).

O segundo item da barra de ferramentas é voltado ao treinamento muscular, intitulado de Meu Movimento (conferir Figura 15). Esta parte do aplicativo dedica-se a trazer informações dos exercícios a serem executados pelo aluno.

Figura 15 – Tela Meu Movimento



(a) Página Inicial do Meu movimento (b) Lista de Exercícios (c) Página de Exercício

Fonte: Selfit (s.d.)

Através desta tela (figura a), aberta ao selecionar o segundo item da barra de ferramentas, é possível visualizar os programas de treinamento (também chamados de fichas de exercício) que podem ser cadastrados ou editados tanto pelo aluno quanto pelo profissional de Educação Física filiado à empresa.

Pode ser cadastrada mais de uma ficha, sendo elas ordenadas por data de cadastro de tal forma que, encabeçando a lista, apresenta-se a primeira ficha da lista ainda não executada. Para visualizar os exercícios presentes nela, é necessário clicar no treino.

Ao clicar no treino (figura b), apresentam-se as informações gerais do treinamento; a opção para emparelhamento da cinta cardíaca ou, caso já esteja emparelhado, o nome da cinta conectada; a lista de exercícios presente no programa de treinamento atual; e o botão para iniciar o treino dentro das dependências da academia selecionada previamente na tela inicial. Também é possível adicionar novos exercícios através do



botão superior direito, sendo retornada uma lista de todos os exercícios agrupados por grupo muscular envolvido.

Selecionando um exercício (figura c), podem ser visualizadas informações a respeito da execução dele, podendo ser iniciado (através do botão iniciar) o treino a partir do exercício selecionado, mesmo que este não seja o primeiro da lista. Após a iniciação do treinamento, o botão Iniciar é trocado pelo Finalizar Exercício, que insere o exercício no histórico de exercícios realizados.

Através da Tela de Resultados, podemos analisar o desempenho do usuário, o que serve como histórico de diversos parâmetros utilizados pelos profissionais da saúde ou da Educação Física para acompanhamento da saúde do aluno (conferir Figura 16).

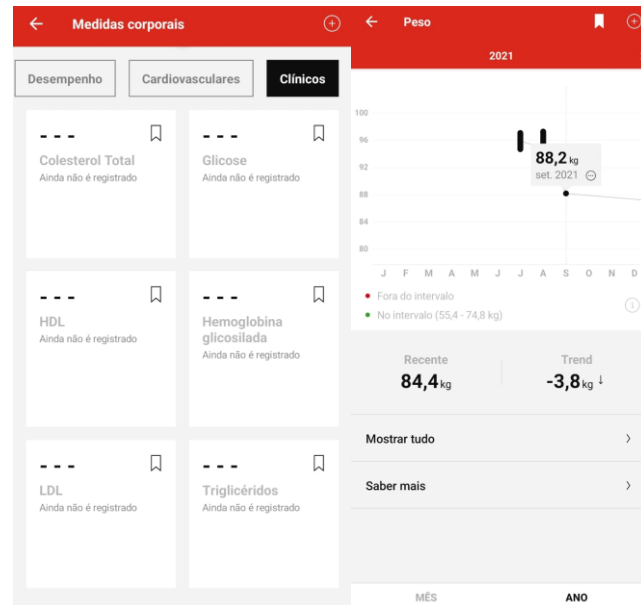
Figura 16 – Tela Resultados



Fonte: Selfit (s.d.)

Nesta tela é possível acessar as outras telas de Resultado e visualizar os gráficos gerados da progressão do aluno (conferir Figura 17).

Figura 17 – Tela Resultados de Medida Corporal



(a) Visualização geral (b) Visualização Específica

Fonte: Selfit (s.d.)

Ao selecionar a opção Medição Corporal, presente na figura 16, o usuário é redirecionado para a tela desta figura: Resultados de Medida Corporal. O primeiro passo deste fluxo ocorre através da visualização geral (figura a), em que é possível visualizar as diversas formas de medida para acompanhamento da saúde do aluno, sendo tais medidas agrupadas por tipo de medição. Então, ao selecionar um tipo de medida, abre-se a tela de visualização específica (figura b), na qual se pode visualizar um gráfico com o histórico das medições, favoritar a medição para facilitar o encontro da página e adicionar uma nova medição para atualização do histórico (conferir Figura 18).

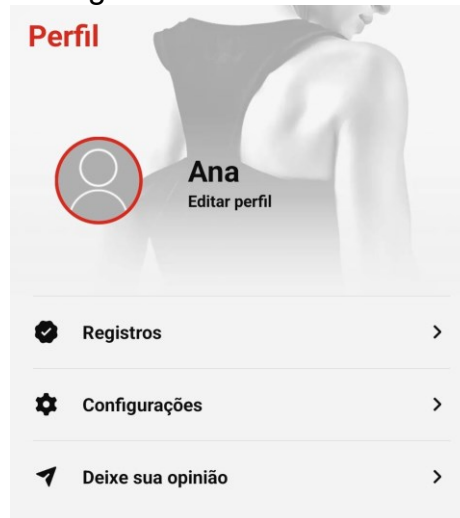
Figura 18 – Tela Resultado de Recordes



Fonte: Selfit (s.d.)

Ao selecionar a opção Registros, presente na figura 16, o usuário é redirecionado para a tela desta figura, ou seja, aquela que aponta o Resultado de Recordes, uma gameificação realizada pela empresa para estimular o aluno na realização da atividade física. Nesta tela, é possível visualizar todos os recordes obtidos e quais podem ainda ser alcançados pelo aluno (conferir Figura 19).

Figura 19 – Tela de Perfil



Fonte: Selfit (s.d.)

Através da tela de Perfil, acessada pela última opção do Menu, é possível personalizar o usuário, visualizar a página de recordes pessoais por meio da opção registros, acessar o menu de configurações do aplicativo e realizar uma avaliação da empresa.

### 3 PROPOSTA E METODOLOGIA

Neste capítulo, serão apresentados os detalhes da metodologia utilizada no desenvolvimento do software proposto, bem como os procedimentos adotados para alcançar os objetivos específicos do projeto.

O software em questão tem como finalidade realizar o controle de atendimento, o controle de uso de aparelhos da academia e o controle de alunos em academias voltadas ao treinamento personalizado de pequeno e médio porte. Para alcançar esses objetivos, é essencial uma abordagem metodológica eficiente que guie todas as etapas do desenvolvimento.

#### 3.1 METODOLOGIA

O tipo de pesquisa será de campo, executando os conteúdos aprendidos durante o Curso de Tecnologia da Informação e da Comunicação. Tendo em vista que esta é uma pesquisa experimental, os procedimentos de testagem são as verificações de qualidade do software e entrevistas com profissionais da área após o período de teste, para identificar se os requisitos foram satisfeitos.

O projeto será desenvolvido através do conteúdo explicitado nas aulas de Desenvolvimento de Sistemas Web, Engenharia de Software 1 e 2 do Curso de TIC e também através de pesquisas, lançando mão de tutoriais e artigos na Internet e leitura de livros relacionados ao desenvolvimento de software e à linguagem Python voltada a Framework Django, HTML, CSS e orientação a objeto.

O método que será utilizado no desenvolvimento do software é a metodologia Ágil SCRUM, sendo guiada por objetivos a partir da criação dos documentos de especificação do software, os objetivos são separados em estruturais (como estruturação das conexões necessárias, telas, funções úteis para todo sistema etc.) e objetivos específicos (como os casos de uso, fluxos dos dados etc.).

O quadro metodológico segue os seguintes passos:

- Etapa 1 (Análise de Requisitos): Na primeira etapa, é estabelecido o requisito para todos os elementos do sistema, levando em consideração os requisitos levantados por meio de entrevistas. Esta visão do sistema é fundamental para o software, assim como para a função, o desempenho e a interface exigidos;
- Etapa 2 (Modelagem): concentra-se em especificar os casos de uso, framework, estruturação das telas. Esta etapa se resume em construir, a partir dos requisitos, um guia para que os programadores possam iniciar o desenvolvimento;
- Etapa 3 (codificação): a etapa da codificação executa a tarefa de traduzir o projeto para uma forma legível para a máquina. Se o projeto for executado detalhadamente, a codificação pode ser executada mecanicamente. Esta etapa engloba a etapa de testes;
- Etapa 4 (Utilização Beta): esta etapa concentra-se na usabilidade do sistema, entregando acessos para usuários específicos, de modo que: validem a utilização do software; tragam melhorias ao sistema; e encontrem erros/falhas;
- Etapa 5 (Manutenção): depois da etapa 4, ele sofrerá as alterações necessárias tanto para os erros que foram encontrados quanto para a adaptação do software, a fim de acomodar mudanças em seu ambiente externo ou ainda devido às exigências do cliente em acréscimos funcionais ou de desempenho.
- Etapa 6 (Gestão de Qualidade): sendo a última etapa deste projeto, o objetivo é verificar se o software cumpre seu objetivo, buscando possíveis usuários para utilizar a aplicação por determinado tempo e, após isso, pedindo-lhes para responder questionários a respeito de sua experiência.

## 4 MODELAGEM E IMPLEMENTAÇÃO

Esta seção busca delimitar as características do software a ser desenvolvido, parte imprescindível para que se crie um objeto único de comunicação entre todos os envolvidos do projeto. Utilizando os conceitos presentes na Engenharia de Software, foi desenvolvido um Documento de Especificação de Requisitos para ser apresentado neste capítulo, tendo em vista que

O documento de requisitos de um software contém todos os requisitos funcionais e de qualidade do software, incluindo as capacidades do produto, os recursos disponíveis, os benefícios e os critérios de aceitação (TURINE, 1996, p. 7).

### 4.1 PROPÓSITO

Com o objetivo de gerenciar o uso de máquinas de academia, dentro do contexto do treinamento personalizado realizado por múltiplos treinadores paralelamente, este software tem por foco melhorar a comunicação entre treinadores, oferecendo informações das máquinas livres e em uso. Para os alunos, o aplicativo serve como lembrete do exercício atual, trazendo informações de uso.

### 4.2 ESCOPO

A AcadApp é uma aplicação salva na nuvem que irá substituir a comunicação verbal no uso de máquinas de academia entre treinadores, além de substituir a agenda para a realização de agendamento de atendimentos. Já para o aluno, o aplicativo servirá para visualização dos seus exercícios e avaliação do atendimento.

## 4.3 DEFINIÇÃO DO PRODUTO

### 4.3.1 Perspectiva do Produto

Em uma sociedade em que o interesse pelo bem-estar e saúde resulta em uma grande quantidade de alunos em busca de treinadores personalizados, a AcadApp vem como um comunicador entre treinadores de uma mesma academia.

### 4.3.2 Funções do Produto

Suas principais funções envolvem:

- **Cadastro de atendimento:** salva, em servidor na nuvem, o horário em que determinado aluno será atendido pelo treinador;
- **Encaminhar Aluno:** salva, em servidor na nuvem, o uso da máquina em determinado atendimento, alterando o status da máquina e do aluno.

### 4.3.3 Características do Usuário

As classes de usuário necessárias para a construção do software são:

- Treinador
  - Frequência de uso: utilizar para cadastrar atendimentos e adicionar uso de máquinas (exercícios) neles;
  - Subconjunto de classes utilizadas: Atendimento, Máquina, Aluno;
  - Conhecimento técnico: Usuário padrão alto (necessita de treinamento para utilização da aplicação);
  - Nível de segurança: Médio;
  - Nível de conhecimento: Graduação completa em Educação Física ou associados.



- **Treinador Master**
  - **Frequência de uso:** Cadastrar máquinas e treinadores;
  - **Subconjunto de classes utilizadas:** Treinador, Máquina;
  - **Conhecimento técnico:** Usuário padrão alto (necessita de treinamento para utilização da aplicação);
  - **Nível de segurança:** Alto;
  - **Nível de conhecimento:** Alfabetização.
- **Administrador**
  - **Frequência de uso:** Cadastrar academias e treinadores Master;
  - **Subconjunto de classes utilizadas:** Academia, Treinador Master;
  - **Conhecimento técnico:** Usuário padrão;
  - **Nível de segurança:** Alto;
  - **Nível de conhecimento:** Alfabetização.
- **Aluno**
  - **Frequência de uso:** Visualizar exercício e avaliar atendimento;
  - **Subconjunto de classes utilizadas:** Atendimento;
  - **Conhecimento técnico:** Usuário padrão;
  - **Nível de segurança:** Baixo;
  - **Nível de conhecimento:** Alfabetização.

#### 4.3.4 Limitações

As principais limitações, que relativas aos Usuários Administrador e Treinador Master, são:

1. Academia para ser cadastrada necessita de um CNPJ válido;
2. Os treinadores, para serem cadastrados, necessitam de ter inscrição no Confef ou no Cref válido, além de um CPF válido.

### **4.3.5 Definições**

Exercício: forma de utilização de determinada máquina;

Atendimento Geral: horário combinado de encontro entre Aluno e Treinador;

Atendimento Personalizado: atendimento realizado em dia e horário específicos.

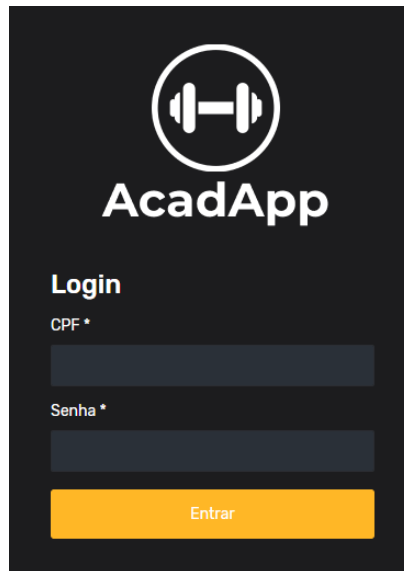
## **4.4 IMPLEMENTAÇÃO**

### **4.4.1 Interface Externa**

#### *4.4.1.1 Tela de Login*

Nesta parte do sistema, teremos uma estrutura seguindo a ideia minimalista, removendo desperdícios e informações desnecessárias, de modo a criar algo visualmente fácil a qualquer usuário. As cores utilizadas tanto pela tela de login quanto por todo o sistema são tons de cinza e amarelo (conferir Figura 20).

Figura 20 – Modelo estrutural de tela de login

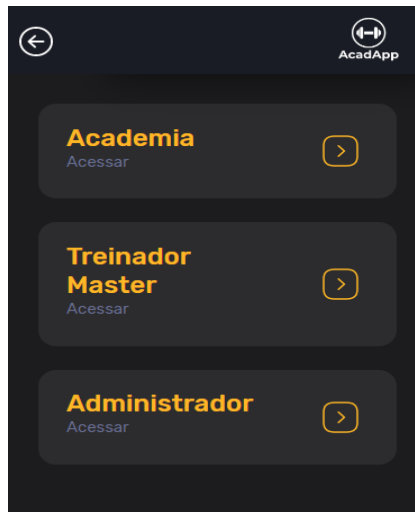


Fonte: elaborado pela autora

#### 4.4.1.2 Tela de Menu

Após realizar Login, a primeira tela apresentada é o Menu. A proposta minimalista do design do projeto foi importante para a visualização rápida através de celulares Smartphones, havendo, porém, a necessidade de ajuste ao tamanho limitado de tela sem que houvesse excesso de informações, conforme abaixo (conferir Figura 21).

Figura 21 – Modelo estrutural da tela de Menu

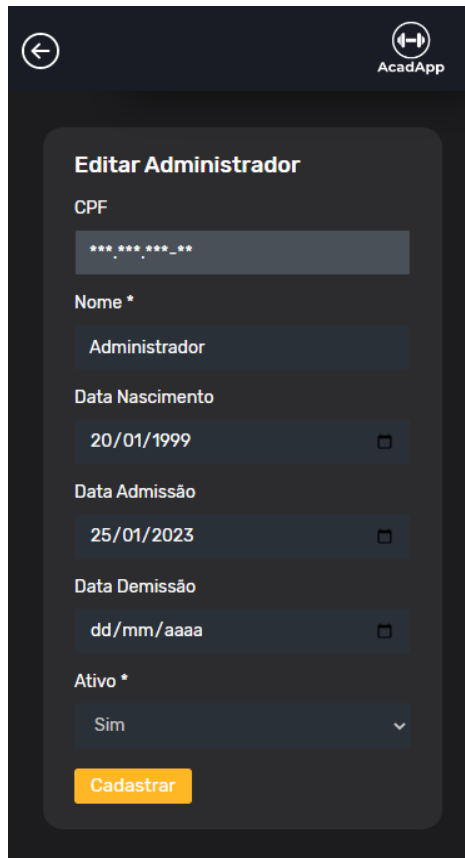


Fonte: elaborado pela autora

#### 4.4.1.3 Tela de Cadastro

Seguindo o conceito, a tela de Cadastro deve dispor um formulário simples, com identificadores sucintos e objetivos. Em caso de campo obrigatório, um asterisco é colocado em conjunto, dentro das caixas de inserção, sem valor. Os identificadores devem ser repetidos. O ponto mais importante aqui é o tamanho acessível para qualquer tipo de tela de dispositivos móveis (conferir Figura 22).

Figura 22 – Modelo estrutural da tela de Cadastro



A imagem mostra a interface de usuário para a tela de cadastro de um administrador. O formulário é intitulado "Editar Administrador" e contém os seguintes campos:

- CPF: Campo com máscara de entrada (\*\*\* \*\*\*)\_\*\*).
- Nome \*: Campo com o valor "Administrador".
- Data Nascimento: Campo com o valor "20/01/1999" e ícone de calendário.
- Data Admissão: Campo com o valor "25/01/2023" e ícone de calendário.
- Data Demissão: Campo com o valor "dd/mm/aaaa" e ícone de calendário.
- Ativo \*: Campo com o valor "Sim" e ícone de seta para baixo.

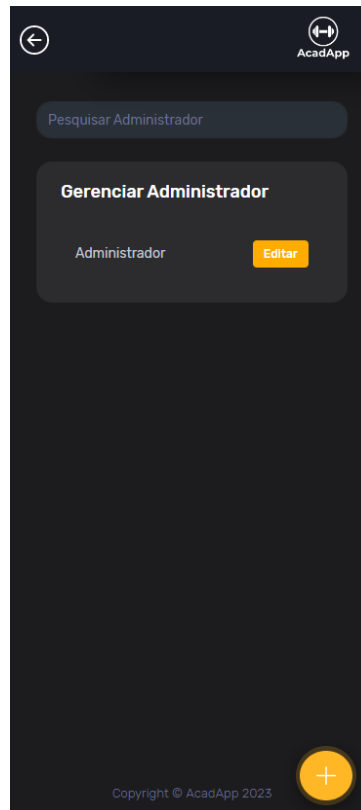
Um botão amarelo "Cadastrar" está localizado na base do formulário. No topo da tela, há um ícone de seta para trás e o logotipo "AcadApp".

Fonte: elaborado pela autora

#### 4.4.1.4 Tela de Gerenciamento

As telas de visualização geral/gerenciamento apresentarão tabelas com apenas as informações gerais necessárias, de tal maneira que, ao clicar em uma linha, será possível escolher ações correspondentes à visualização escolhida. Assim como no topo, haverá as ações da tabela opcionais, como filtros, e, em cantos estratégicos, serão dispostas as ações mais importantes para o sistema, como o botão de Voltar e o de Cadastro, conforme abaixo (conferir Figura 23).

Figura 23 – Modelo estrutural da tela de Gerenciamento



Fonte: elaborado pela autora

## 4.4.2 Funções

### 4.4.2.1 Requisitos do Usuário

#### 4.4.2.1.1 Requisitos Funcionais

1. Restringir acesso com login e senha;
2. Criar sistema de permissão (Administrador/Treinador/Treinador Master/Aluno);
3. Realizar o cadastro do horário do atendimento do aluno, caso possua permissão Treinador;
4. Realizar cadastro de máquinas e treinadores a partir da permissão Treinador Master;
5. Cadastrar exercícios dentro dos atendimentos a partir da permissão Treinador;
6. Administradores cadastram os treinadores Master e as academias;

#### 4.4.2.1.2 Regras de Negócio

1. Cada Academia deve ter uma estrutura Própria (RF1);
2. A academia possui máquinas (RF4);
3. A academia possui treinadores (RF4);
4. Cada tipo de usuário deve ter um painel próprio (RF2);
5. Os alunos possuem apenas um atendimento por dia com um treinador (RF3);
6. Os Treinadores realizam um acordo informal de horários de Atendimento com o Aluno e o Cadastra no sistema (RF3);
7. Os treinadores designam seus alunos nos exercícios que são associados às máquinas (RF5);
8. O sistema deve aceitar um CPF (não editável) por usuário;
9. O sistema deve aceitar um CNPJ (não editável) por academia.
10. O sistema deve notificar os alunos que iniciarão no próximo horário para os treinadores;

#### 4.4.2.1.3 Requisitos Não Funcionais

1. Voltado ao uso de celulares/tablets;
2. Páginas responsivas e intuitivas;
3. Tempo máximo de carregamento de cinco segundos;

#### 4.4.2.1.4 Ferramentas Utilizadas

1. Google Drive
  - a. Repositório de Conteúdo
  - b. Organização de cronograma
  - c. Criação de Formulários
2. Creately
  - a. Diagrama de Casos de Uso
3. Notion
  - a. Organização das tarefas da Sprint
4. BrModelo
  - a. Criação do Modelo Lógico de Dados
5. Figma

- a. Modelagem da Interface
- b. Prototipagem
- 6. PyCharm
  - a. Ambiente de Desenvolvimento
- 7. FreePik
  - a. Ícones
- 8. Django
  - a. Framework de Desenvolvimento
- 9. Python
  - a. Linguagem de Desenvolvimento
- 10. Registro.br
  - a. Compra de Domínio
- 11. PythonAnywhere
  - a. Hospedagem Web

#### 4.4.2.2 Diagrama de Caso de Uso

Figura 24 – Diagrama de caso de uso



Fonte: elaborado pela autora



#### 4.4.2.3 Descrição dos Casos de Uso

##### 4.4.2.3.1 Administrador

Tabela 1 – Fluxos Realizados pelo Administrador

UC01	Fluxo Gerenciamento Academia
<b>Descrição</b>	Permite o gerenciamento das academias cadastradas
<b>Ator Principal</b>	Administrador
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de administrador
<b>Pós Condições</b>	Visualização de relatório de academias
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Academia no menu.	
	2. Retorna todas as Academias presentes no sistema em ordem alfabética. Opções Gerais: Cadastrar/ Voltar. Opções de cada cadastro: Editar.
3. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
3.1.1. Seleciona a opção Cadastrar.	
	3.1.2. Retorna à página com formulário de Cadastro de Academia. Com apenas o input para o CNPJ.
3.1.3. Digita o CNPJ da Academia e aperta no botão Verificar.	
	3.1.4. Valida CNPJ novo. Retorna os outros inputs vazios e opções: Salvar / Cancelar.
3.1.5. Preenche o formulário e confirma a inserção dos dados.	
	3.1.6. Retorna a mensagem: Academia Cadastrada com Sucesso! Gostaria de cadastrar um treinador master para a nova academia? Opções: Sim / Não.
3.1.7. Seleciona a opção Sim.	
	3.1.8. Abre fluxo Gerenciamento Treinador Master com a

	Academia pré-selecionada e não mutável.
<b>Fluxo Alternativo 02</b>	
3.2.1. Seleciona a opção Editar em uma Academia.	
	3.2.2. Abre o formulário de Fluxo de Alternativo 01 com as informações carregadas do cadastro.
<b>Fluxo Alternativo 03</b>	
	3.1.4.3.1. Verifica cadastro já existente. Retorna mensagem: Academia já cadastrada, modo edição. Retorna formulário com inputs preenchidos do cadastro e opções: Salvar / Deletar / Voltar.
	3.1.4.3.2. Retorna ao passo 3.1.5.
<b>Fluxo Alternativo 04</b>	
	3.1.4.4.1. Verifica CNPJ inválido. Retorna mensagem: CNPJ inválido, tente novamente com outro CNPJ.
	3.1.4.4.2. Retorna ao passo 3.1.2.
<b>Fluxo Alternativo 05</b>	
3.1.5.5.1. Preenche formulário e cancela inserção dos dados.	
	3.1.5.5.2. Retorna ao passo 2.
<b>Fluxo Alternativo 06</b>	
3.1.7.6.1. Seleciona a opção Não.	
	3.7.1.6.2. Retorna ao Menu inicial.
<b>Fluxo Alternativo 07</b>	
3.7.1. Seleciona a opção Voltar.	
	3.7.2. Retorna ao Menu inicial.

UC02	Fluxo Gerenciamento Treinador Master
<b>Descrição</b>	Permite o gerenciamento dos Treinadores Master cadastrados
<b>Ator Principal</b>	Administrador
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de administrador

<b>Pós Condições</b>	Visualização de relatório de treinadores master
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Treinador Master no menu.	
	2. Retorna todos os Treinadores Master presentes no sistema em ordem alfabética. Opções Gerais: Cadastrar/ Voltar. Opções de cada cadastro: Editar.
3. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
3.1.1. Seleciona a opção Cadastrar.	
	3.1.2. Retorna à página com formulário de Cadastro de Treinador Master. Com apenas os inputs para: CPF/Email/Senha.
3.1.3. Preenche formulário e submete os dados.	
	3.1.4. Traz outros inputs do formulário.
3.1.5. Preenche o formulário e confirma a inserção dos dados.	
	3.1.6. Retorna a mensagem: Treinador Master Cadastrado com Sucesso!
	3.1.7. Retorna ao passo 2.
<b>Fluxo Alternativo 02</b>	
3.2.1. Seleciona a opção Editar em um Treinador Master.	
	3.2.2. Abre formulário de Fluxo de Alternativo 01 com as informações carregadas do cadastro e com a opção Deletar.
<b>Fluxo Alternativo 03</b>	
	3.1.4.3.1. Verifica cadastro já existente. Retorna mensagem: Treinador Master já cadastrado, modo edição. Retorna formulário com inputs preenchidos do cadastro e opções: Salvar / Deletar / Voltar.
	3.1.4.3.2. Retorna ao passo 3.1.5.
<b>Fluxo Alternativo 04</b>	

	3.1.4.4.1. Verifica CPF inválido. Retorna mensagem: CPF inválido, tente novamente com outro CPF.
	3.1.4.4.2. Retorna ao passo 3.1.2.
<b>Fluxo Alternativo 05</b>	
3.1.5.5.1. Preenche o formulário e aperta o botão Voltar.	
	3.1.5.5.2. Retorna ao passo 2.
<b>Fluxo Alternativo 06</b>	
3.6.1. Seleciona a opção Voltar.	
	3.6.2. Retorna ao Menu inicial.

<b>UC03</b>	<b>Fluxo Gerenciamento Administrador</b>
<b>Descrição</b>	Permite o gerenciamento dos administradores cadastrados
<b>Ator Principal</b>	Administrador
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de administrador
<b>Pós Condições</b>	Visualização de relatório de administrador
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Administrador no menu.	
	2. Retorna todos Administradores presentes no sistema em ordem alfabética. Opções Gerais: Cadastrar/ Voltar/Filtrar. Opções de cada cadastro: Editar.
3. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
3.1.1. Seleciona a opção Cadastrar.	
	3.1.2. Retorna à página com formulário de Cadastro de Administrador. Com apenas os inputs: CPF/Email/Senha.
3.1.3. Insere e submete os dados.	
	3.1.4. Valida como um CPF novo no sistema. Retorna os outros inputs vazios e opções: Cadastrar / Cancelar.

3.1.5. Preenche o formulário e confirma a inserção dos dados.	
	3.1.6. Retorna a mensagem: Administrador Cadastrado com Sucesso!
	3.1.7. Retorna passo 2.
<b>Fluxo Alternativo 02</b>	
3.2.1. Seleciona a opção Editar em um Cadastro.	
	3.2.2. Abre o formulário de Fluxo de Alternativo 01 com as informações carregadas do cadastro.
<b>Fluxo Alternativo 03</b>	
	3.1.4.3.1. Verifica cadastro já existente. Retorna mensagem: Administrador já cadastrado, modo edição. Retorna formulário com inputs preenchidos do cadastro e opções: Salvar / Deletar / Voltar.
	3.1.4.3.2. Retorna ao passo 3.1.5.
<b>Fluxo Alternativo 04</b>	
	3.1.4.4.1. Verifica CPF inválido. Retorna mensagem: CPF inválido, tente novamente com outro CPF.
	3.1.4.4.2. Retorna ao passo 3.1.2.
<b>Fluxo Alternativo 05</b>	
3.1.5.5.1. Preenche formulário e cancela inserção dos dados.	
	3.1.5.5.2. Retorna ao passo 2.
<b>Fluxo Alternativo 06</b>	
3.6.1. Seleciona a opção Voltar.	
	3.6.2. Retorna ao Menu inicial.

Fonte: elaborado pela autora

#### 4.4.2.3.2 Treinador Master

Tabela 2 – Fluxos Realizados pelo Treinador Master

UC04	Fluxo Gerenciamento de Máquina
Descrição	Permite o gerenciamento das máquinas cadastradas da

	academia
<b>Ator Principal</b>	Treinador Master
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de Treinador Master
<b>Pós Condições</b>	Visualização de relatório de máquinas
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Máquina no menu.	
	2. Retorna todas as Máquinas presentes na Academia logada em ordem alfabética. Cadastrar/ Voltar. Opções de cada cadastro: Editar.
3. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
3.1.1. Seleciona a opção Cadastrar.	
	3.1.2. Retorna à página com formulário de Cadastro de Máquina.
3.1.3. Preenche o formulário e confirma a inserção dos dados.	
	3.1.4. Retorna a mensagem: Máquina Cadastrada com Sucesso! Retorna ao passo 2.
<b>Fluxo Alternativo 02</b>	
3.2.1. Seleciona a opção Editar em uma Máquina.	
	3.2.2. Abre formulário de Fluxo de Cadastrar Máquina com as informações carregadas do cadastro
<b>Fluxo Alternativo 03</b>	
3.3.1. Seleciona a opção Voltar.	
	3.3.2. Retorna ao Menu.

<b>UC05</b>	<b>Fluxo Gerenciamento Treinador</b>
<b>Descrição</b>	Permite o gerenciamento dos Treinadores cadastrados na academia logada

<b>Ator Principal</b>	Treinador Master
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de treinador Master
<b>Pós Condições</b>	Visualização de relatório de treinadores
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Treinador no menu.	
	2. Retorna todos os Treinadores presentes no sistema em ordem alfabética. Opções Gerais: Cadastrar/ Voltar. Opções de cada cadastro: Editar.
3. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
3.1.1. Seleciona a opção Cadastrar.	
	3.1.2. Retorna à página com formulário de Cadastro de Treinador. Com apenas o input para o CPF/Email/Senha.
3.1.3. Digita CPF/Email/Senha do Treinador e submete os dados.	
	3.1.4. Verifica CPF novo. Retorna os outros inputs vazios e opções: Salvar / Cancelar.
3.1.5. Preenche o formulário e confirma a inserção dos dados.	
	3.1.6. Retorna a mensagem: Treinador Cadastrado com Sucesso!
	3.1.7. Retorna ao passo 2.
<b>Fluxo Alternativo 02</b>	
3.2.1. Seleciona a opção Editar em um Treinador.	
	3.2.2. Abre o formulário do Fluxo Alternativo 1 com as informações carregadas do cadastro.
<b>Fluxo Alternativo 03</b>	
	3.1.4.3.1. Verifica cadastro já existente. Retorna mensagem: Treinador já cadastrado, modo edição. Retorna formulário com inputs preenchidos do cadastro e opções: Salvar / Cancelar.
	3.1.4.3.2. Retorna ao passo 3.1.5.

<b>Fluxo Alternativo 04</b>	
	3.1.4.4.1. Verifica CPF inválido. Retorna mensagem: CPF inválido, tente novamente com outro CPF.
	3.1.4.4.2. Retorna ao passo 3.1.2.
<b>Fluxo Alternativo 05</b>	
3.1.5.5.1. Preenche o formulário e aperta em Voltar.	
	3.1.5.5.2. Retorna ao passo 2.
<b>Fluxo Alternativo 06</b>	
3.6.1. Seleciona a opção Voltar.	
	3.6.2. Retorna ao Menu.

<b>UC06</b>	<b>Fluxo Gerenciamento de Grupo Muscular</b>
<b>Descrição</b>	Permite o gerenciamento dos grupos musculares cadastrados na academia.
<b>Ator Principal</b>	Treinador Master
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de Treinador Master
<b>Pós Condições</b>	Visualização de relatório de Grupo Muscular
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Grupo Muscular no menu.	
	2. Retorna todos os Grupos Musculares cadastrados na Academia logada em ordem alfabética com a quantidade de exercícios que os utiliza. Opções Gerais: Voltar.

<b>UC07</b>	<b>Fluxo Gerenciamento de Avaliação</b>
<b>Descrição</b>	Permite o gerenciamento das Avaliações dos Atendimentos cadastrados na academia
<b>Ator Principal</b>	Treinador Master



<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de Treinador Master
<b>Pós Condições</b>	Visualização de relatório de Avaliação
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Avaliação no menu.	
	2. Retorna todas as Avaliações presentes na Academia logada ordenadas por Data em ordem decrescente. Opções Gerais: Voltar. Opções de cada cadastro: Visualizar.
3. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
3.1.1. Seleciona a opção Visualizar em uma avaliação.	
	3.1.2. Retorna à Página com os dados da Avaliação.
<b>Fluxo Alternativo 02</b>	
3.2.1. Seleciona a opção Voltar.	
	3.2.2. Retorna ao Menu.

Fonte: elaborado pela autora

#### 4.4.2.3.3 Treinador

Tabela 3 – Fluxos Realizados pelo Treinador

<b>UC08</b>	<b>Fluxo Gerenciamento Aluno</b>
<b>Descrição</b>	Permite o gerenciamento dos Alunos do Treinador cadastrados na academia logada
<b>Ator Principal</b>	Treinador
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de treinador
<b>Pós Condições</b>	Visualização de relatório de alunos
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>

1. Seleciona a opção Aluno no menu.	
	2. Retorna todos os Alunos presentes no sistema em ordem alfabética. Opções Gerais: Cadastrar/ Voltar. Opções de cada cadastro: Editar.
3. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
3.1.1. Seleciona a opção Cadastrar.	
	3.1.2. Retorna à página com formulário de Cadastro de Aluno. Com apenas o input para o CPF / Email / Senha.
3.1.3. Digita CPF / E-mail / Senha do Aluno e submete os dados.	
	3.1.4. Verifica CPF novo. Retorna os outros inputs vazios e opções: Salvar / Cancelar.
3.1.5. Preenche o formulário e confirma a inserção dos dados.	
	3.1.6. Cadastra novo usuário de permissão Aluno. Retorna ao Fluxo Gerenciar Atendimento no passo 3.5.2.
<b>Fluxo Alternativo 02</b>	
3.2.1. Seleciona a opção Editar em um Aluno.	
	3.2.2. Abre o formulário de Fluxo Alternativo 1 com as informações carregadas do cadastro.
<b>Fluxo Alternativo 03</b>	
	3.1.4.3.1. Verifica cadastro já existente. Retorna mensagem: Aluno já cadastrado, modo edição. Retorna formulário com inputs preenchidos do cadastro e opções: Salvar / Voltar / Deletar.
	3.1.4.3.2. Retorna ao passo 3.1.5.
<b>Fluxo Alternativo 04</b>	
	3.1.4.4.1. Verifica CPF inválido. Retorna mensagem: CPF inválido, tente novamente com outro CPF.
	3.1.4.4.2. Retorna ao passo 3.1.2.
<b>Fluxo Alternativo 05</b>	
3.1.5.5.1. Preenche formulário e cancela inserção dos dados.	

	3.1.5.5.2. Retorna ao passo 2.
<b>Fluxo Alternativo 06</b>	
3.6.1. Selecione a opção Voltar.	
	3.6.2. Retorna ao Menu.
<b>Fluxo Alternativo 07</b>	
3.1.7.1 Seleciona Opção: Passar Aluno.	
	3.1.7.2. Retorna ao formulário Passar Aluno.
3.1.7.3. Seleciona Treinador e Alunos para serem passados.	
	3.1.7.4 O Atendimento do Aluno vai alterar para o Treinador selecionado. Retorna passo 3.1.7.2. com mensagem de Sucesso

<b>UC09</b>	<b>Fluxo Gerenciar Exercícios</b>
<b>Descrição</b>	Permite o gerenciamento dos Exercícios associados às Maquinas
<b>Ator Principal</b>	Treinador
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de treinador
<b>Pós Condições</b>	Novo Exercício cadastrado no sistema, associado à Máquina
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Exercícios no menu.	
	2. Retorna à página todas as Máquinas cadastradas na Academia, Opções Gerais: Filtrar/ Voltar. Opções de cada cadastro: Exercício.
3. Seleciona a opção Exercício de uma Máquina.	
	4. Retorna à página com todos os Exercícios cadastrados para essa máquina. Botões: Adicionar Exercício / Voltar. Opções de cada cadastro: Editar.
5. Aperta no botão Adicionar Exercício.	

	6. Retorna à página com formulário de Cadastro de Exercício.
7. Preenche o formulário e confirma a inserção dos dados.	
	6. Retorna a mensagem: Exercício Cadastrado com Sucesso! Retorna ao passo 4.
<b>Fluxo Alternativo 01</b>	
3.1.1. Preenche o input de filtro.	
	3.1.2. Após 2 segundos de preenchimento do input, retorna os cadastros que contenham o nome inserido.
<b>Fluxo Alternativo 02</b>	
3.2.1. Aperta o botão Voltar.	
	3.2.2. Retorna ao menu.
<b>Fluxo Alternativo 03</b>	
5.3.1. Aperta o botão Voltar.	
	5.3.2. Retorna ao passo 2.
3.3.1. Aperta o botão Voltar.	
	5.3.2. Retorna ao menu.
<b>Fluxo Alternativo 04</b>	
5.4.1. Aperta o botão Voltar.	
	3.4.1. Retorna ao passo 2.

<b>UC10</b>	<b>Fluxo Gerenciamento de Atendimentos do Treinador</b>
<b>Descrição</b>	Permite o gerenciamento das Atendimentos cadastradas do Treinador
<b>Ator Principal</b>	Treinador
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de Treinador
<b>Pós Condições</b>	Visualização de relatório de Atendimentos
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Atendimento no menu.	

	2. Retorna todos os Alunos do Treinador logado, ordenado por Hora (de atendimento) e Pré filtrado pelo dia presente. Opções Gerais: Filtrar/ Alterar Treinador/ Entrar( em Atendimento) / Cadastrar/ Voltar. Opções de cada cadastro: Editar.
3. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
3.1.1. Digita o nome do aluno a ser filtrado.	
	3.1.2. Após 2 segundos o sistema retorna Alunos que contenham os caracteres inseridos.
	3.1.3. Retorna os cadastros que contenham o nome inserido.
<b>Fluxo Alternativo 02</b>	
3.2.1. Seleciona a Data a ser filtrada.	
	3.2.2. Após 2 segundos o sistema retorna os que contenham os caracteres inseridos.
<b>Fluxo Alternativo 03</b>	
3.3.1. Seleciona a opção Alterar Treinador.	
	3.3.2. Retorna passo 2, com os atendimentos do Treinador Selecionado.
<b>Fluxo Alternativo 04</b>	
3.4.1. Selecione a opção Editar em um Atendimento.	
	3.4.2. Retorna à tabela de Atendimento com cada linha sendo um dia da semana. Opção de cada cadastro: Editar. Clicando na opção Editar é possível cadastrar o horário de atendimento do dia da semana.
	3.4.3. Caso tenha um atendimento filtrado, busca o atendimento e retorna formulário para edição.
3.4.4. Insere os dados no formulário e o submete. Após configurar todos os horários, clica em Voltar.	
	3.5.5. Retorna ao passo 2.
<b>Fluxo Alternativo 05</b>	
3.5.1. Seleciona a opção Entrar em Atendimento.	

	3.5.2. Valida existência de Atendimentos cadastrados na hora clicada.
	3.5.3. Abre o menu de realização de atendimento.
3.5.4. Seleciona a opção Encaminhar (Aluno à um exercício).	
	3.5.5. Retorna à página para seleção de alunos livres.
3.5.6. Seleciona Aluno e submete a inserção do dado.	
	3.5.7. Altera o Status do Aluno para Ocupado e redireciona para a Página de Seleção de Máquinas Livres.
3.5.8. Seleciona Máquinas Livres.	
	3.5.9. Retorna os Exercícios cadastrados para essas Máquinas (Uma por vez).
3.5.10. Seleciona o Exercício que o Aluno deve fazer e a quantidade de séries do exercício.	
	3.5.11. Retorna formulário de especificação de série do Exercício.
3.5.12. Insere os dados da série, podendo adicionar quantos séries desejar.	
	3.5.13. Retorna ao menu Realizar Atendimento.
	3.5.14. Adiciona Exercícios ao Histórico do Atendimento com a coluna Realizado igual a Não.
	3.5.15. Remove Máquinas da subpágina de Máquinas Livres e Aluno da subpágina de Alunos Livres.
	3.5.16. Adiciona o Aluno à subpágina de Alunos Ocupados com as Máquinas associadas com botão Finalizar, adiciona a Máquina à subpágina de Máquinas Ocupadas.
	3.5.17. Retorna Mensagem: "Sucesso aluno encaminhado"
<b>Fluxo Alternativo 06</b>	
3.5.6.1, Seleciona Aluno ocupado.	
	3.5.6.2. Retorna formulário de desocupação de aluno.
3.5.6.3. Seleciona o Aluno e submete a inserção do dado.	

	3.5.6.4. Altera a coluna Realizado dos Exercícios no Histórico do Atendimento para Sim.
	3.5.6.5. Adiciona Máquinas à subpágina de Máquinas Livres e Aluno à subpágina de Alunos Livres.
	3.5.6.6. Remove Máquinas da subpágina de Máquinas Ocupadas e Aluno da subpágina de Alunos Ocupados.
	3.5.6.7. Retorna Mensagem: " Sucesso aluno liberado"
<b>Fluxo Alternativo 07</b>	
3.5.7.1. Seleciona Máquinas Ocupadas.	
	3.5.7.2. Retorna à lista de Máquinas Ocupadas.
<b>Fluxo Alternativo 08</b>	
	3.5.2.8.1. Não valida a existência de Atendimentos cadastrados na hora clicada.
	3.5.2.8.2. Retorna à mensagem de Alerta: "Opa! Não achei nenhum atendimento agora, dá uma olhadinha nos seus atendimentos..".
	3.5.2.8.3. Retorna ao passo 2.
<b>Fluxo Alternativo 09</b>	
3.5.9.1. Seleccione Finalizar.	
	3.5.9.2. Retorna subpágina de alunos livres
3.5.9.3. Seleciona aluno a ser liberado e submete a inserção dos dados.	
	3.5.9.4. Altera o status do atendimento para Finalizado. Retorna ao menu do Realizar Exercício.
<b>Fluxo Alternativo 10</b>	
	3.5.1.10.1. Verifica quantidade de atendimentos neste horário maior que o permitido. Retorna mensagem: Horário lotado, tente novamente com outro horário.
	3.5.1.10.2. Retorna ao passo 2.
<b>Fluxo Alternativo 11</b>	
3.11.1. Seleciona a opção Voltar.	
	3.11.2. Retorna ao menu.

Fonte: elaborado pela autora

4.4.2.3.4 Aluno

Tabela 4 – Fluxos Realizados pelo Aluno

UC11	Fluxo Visualizar Exercício
<b>Descrição</b>	Permite a visualização ou a finalização do Exercício com Status aberto do Atendimento atual
<b>Ator Principal</b>	Aluno
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de aluno
<b>Pós Condições</b>	Visualização do Exercício / Alteração do Status do Exercício
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Exercício no menu.	
	2. Retorna à página com lista dos Exercícios com Status Aberto.
3. Seleciona o Exercício que deseja visualizar.	
	4. Retorna à página com informações do Exercício. Opção: Voltar / Finalizar Série/ Finalizar Exercício.
5. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
5.1.1. Seleciona a opção Voltar.	
	3.1.2. Retorna ao passo 2.
<b>Fluxo Alternativo 02</b>	
5.2.1. Seleciona a opção Finalizar Série.	
	5.2.2. Remove a primeira linha da lista de repetições.
	5.2.3. Retorna ao passo 4.
<b>Fluxo Alternativo 03</b>	
5.3.1. Seleciona a opção Finalizar Exercício.	
	5.3.2. Altera o Status do Exercício para Fechado. Altera o Status da Máquina para Livre. Retorna mensagem: "Sucesso Exercício finalizado."

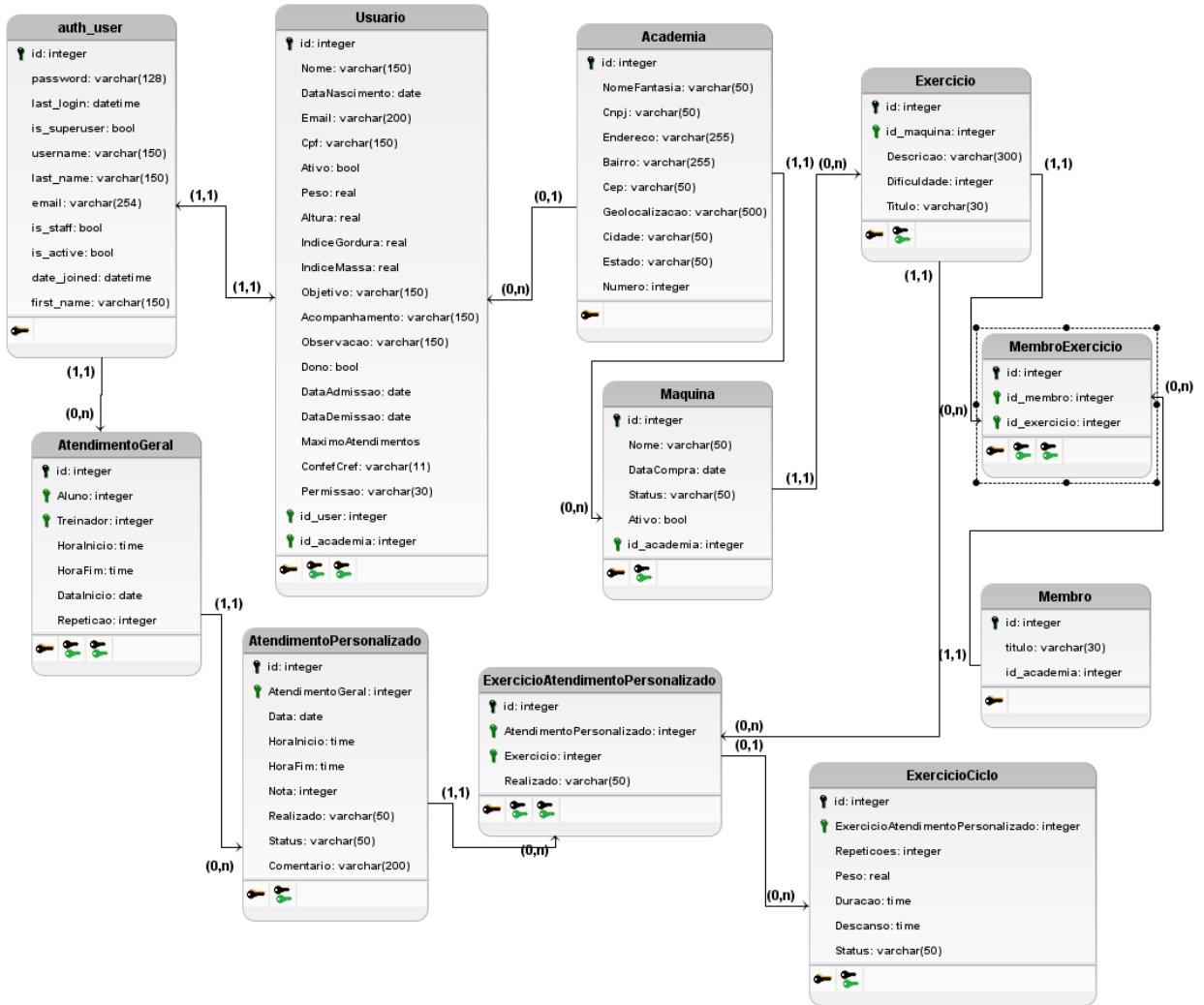


	5.3.3. Retorna passo 2.
<b>UC12</b>	<b>Fluxo Avaliar Atendimento</b>
<b>Descrição</b>	Permite a avaliação dos atendimentos associados ao aluno
<b>Ator Principal</b>	Aluno
<b>Pré-condições</b>	Usuário autenticado no sistema, com permissão de aluno
<b>Pós Condições</b>	Cadastro de Avaliação do Atendimento
<b>Fluxo Principal</b>	
<b>Usuário</b>	<b>Sistema</b>
1. Seleciona a opção Avaliação no menu.	
	2. Retorna todos os Atendimentos presentes no sistema ordenado por Data decrescente. Opções Gerais: Voltar. Opções de cada cadastro: Avaliar.
3. Seleciona a opção desejada.	
<b>Fluxo Alternativo 01</b>	
3.1.1. Seleciona a opção Avaliar em um Atendimento.	
	3.1.2. Abre formulário de Avaliar Atendimento, com visualização dos Exercícios vinculados.
3.1.3. Insere informações e submete os dados.	
	3.1.4. Cadastra avaliação.
	3.1.5. Retorna à mensagem: "Avaliação cadastrada com Sucesso!".
	3.1.6. Retorna ao passo 2.
<b>Fluxo Alternativo 02</b>	
3.2.1. Seleciona a opção Voltar.	
	3.2.2. Retorna ao Menu.

Fonte: elaborado pela autora

4.4.2.4 Modelo Lógico

Figura 25 – Modelo Lógico



Fonte: elaborado pela autora

## **5 RESULTADOS**

Neste capítulo, apresentamos os resultados obtidos na implementação do aplicativo desenvolvido durante a presente pesquisa. Para obtê-los foram realizadas avaliações da usabilidade, medida através do teste de usabilidade e qualidade, e avaliações da eficácia do software, feita por meio do teste Beta. Também será apresentada uma análise do impacto na comunicação e gerenciamento de atendimentos em academias de treinamento personalizado.

### **5.1 TESTE BETA**

Após o desenvolvimento completo do sistema, passamos para a fase de teste beta, cujo objetivo é avaliar se todas as funcionalidades do software estão agindo conforme o esperado, ou seja, se o aplicativo está funcionando da forma descrita no documento de especificação de requisitos. Durante essa etapa, um terceiro realiza a leitura das descrições dos casos de uso e, em seguida, executa todos os fluxos do sistema, após isso, preenche-se um formulário para documentar possíveis problemas encontrados. Essa abordagem nos permite garantir a qualidade do software e identificar falhas antes de sua implantação.

#### **5.1.1 Questionário**

Durante a fase de teste beta, elaboramos um questionário com perguntas estratégicas para avaliar se os fluxos do sistema estavam funcionando corretamente. As perguntas selecionadas foram:

1. Qual fluxo você está testando?
2. Esta é uma tela de qual padrão?
3. Quais passos você realizou neste fluxo?

4. Qual é a sua escala de satisfação com este fluxo?
5. Você tem algum comentário a respeito deste fluxo?
6. Este fluxo é semelhante ao documento de especificação de requisitos?
7. Este fluxo está agradável visualmente?
8. Ocorreu algum erro durante a execução do fluxo?
9. Caso tenha respondido "Sim" ou "Talvez", poderia explicar melhor o que aconteceu e se há alguma dúvida?
10. Se possível, poderia enviar uma foto do erro?
11. Este fluxo foi intuitivo para você?
12. Você tem algum comentário adicional, crítica ou sugestão?

Com essas perguntas, conseguimos identificar possíveis problemas nos fluxos do sistema e coletar um feedback valioso para aprimorar a experiência dos usuários.

### **5.1.2 Análise**

Após duas sessões de testes individuais (feitos por um terceiro sem contato algum com o código ou com o aplicativo anteriormente, utilizando apenas o documento de especificação de requisitos para guiar sua interação com o sistema) e mais uma sessão de teste em grupo (feito com os dois programadores que trabalharam no projeto e o avaliador da fase anterior) foram verificados os casos de uso da documentação e corrigidos erros não identificados anteriormente. Os resultados dos formulários foram sintetizados e cruzados para recolher dados a respeito do sistema. A seguir, apresentamos os dados gerados pelo formulário utilizado nas duas sessões de teste individual e quais melhorias dentro do sistema estas inferências trouxeram.

Através da questão 1, podemos verificar que apenas um fluxo necessitou de uma nova testagem. Isto se deve a dois fatos: primeiro, a maioria dos fluxos necessitam de informações de outro fluxo dentro do sistema (impedindo os próximos fluxos, no caso de

um erro estrutural); e, segundo os testes durante o desenvolvimento foram produtivos o suficiente para que o fluxo passasse de primeira nos testes individuais.

Através da questão 2, podemos verificar que a maioria dos testes foram realizados em páginas do tipo relatório, nas quais podemos ver todos os cadastros de alguma entidade e realizar ações a partir dela, como: deletar, filtrar, realizar um novo cadastro, editar etc. Pode-se imaginar o motivo por trás dessa informação: já que o número de ações possíveis dentro da página de relatório é maior que os demais, estas tornam-se o ponto inicial/final da maioria dos fluxos.

Através da questão 3, podemos identificar uma pergunta voltada à especificação de passos realizados pelo usuário, importante dado para casos de erro, quando se precisa identificar onde está sendo disparado o erro para ajustá-lo.

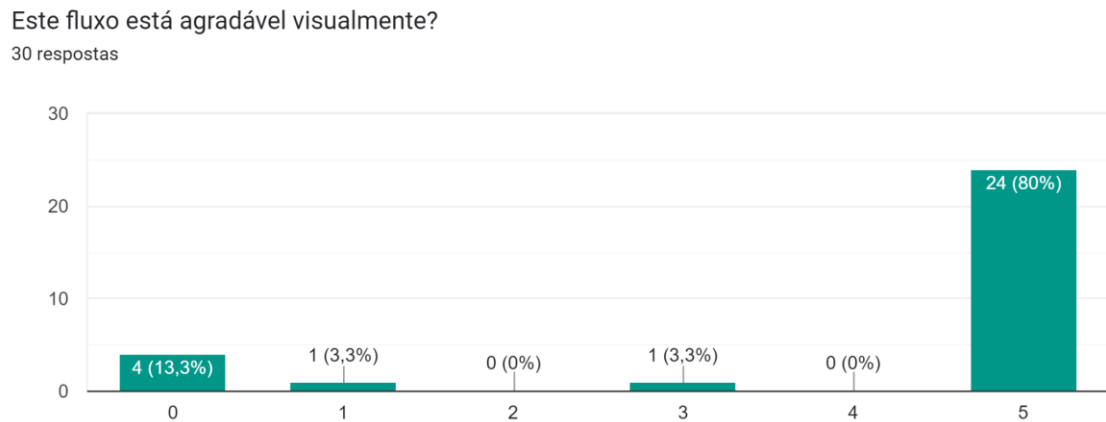
Através da questão 4, podemos facilitar a busca por fluxos incorretos ou disparo de bugs. A satisfação com o fluxo parte de uma característica subjetiva muito relacionada com a facilidade com que o usuário consegue executar os passos dentro do sistema de forma fácil e intuitiva.

Através da questão 5, verificamos pontos que não conseguem ser encaixados em outra questão, como dúvidas, necessidade de reajustar o documento de especificação de requisitos etc.

Através da questão 6, identificamos discrepâncias entre o sistema e o documento de especificação de requisitos que podem ser gerados por erros ou mudanças de fluxo e não atualização do documento. O importante em qualquer um dos casos é identificar o problema e resolvê-lo. Em conjunto com mais algumas questões (8, 9 e 10), é possível realizar uma delimitação melhor do problema.

Através da questão 7, verificamos a necessidade de melhoria na estrutura das telas. Este ponto é importante, pois nesta fase algumas telas ainda estavam em construção, de modo que conseguimos documentar quais as telas que necessitavam de mais atenção. As telas categorizadas com 0 estão totalmente cruas e com apenas um html simples para demarcação de informações a serem colocadas na tela; já as categorizadas com 5 podem-se considerar prontas (conferir Figura 26).

Figura 26 – Este fluxo está agradável visualmente? Respostas

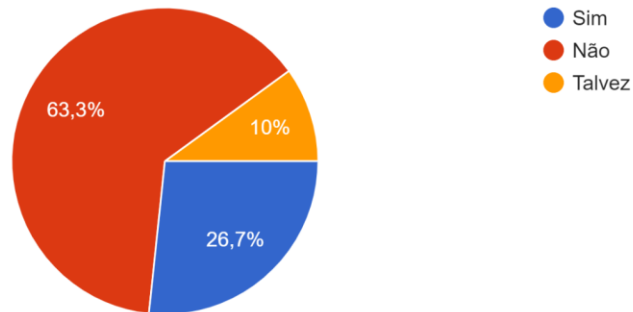


Fonte: elaborado pela autora

Como dito anteriormente, a questão 8 identifica de maneira fácil e rápida fluxos que não estão agindo como deveriam (conferir Figura 27). Um ponto interessante de se ressaltar é que, mesmo com o aplicativo em desenvolvimento, a quantidade de erros/dúvidas é bem baixa (36,7%). Isso se deve a alguns fatores como: documentação suficiente, utilização da documentação para a delimitação de tarefas e extensa quantidade de testes realizados por conta de fluxos cruzados.

Figura 27 – Ocorreu algum erro durante a execução do fluxo? Respostas

Ocorreu algum erro durante a execução do fluxo?  
30 respostas



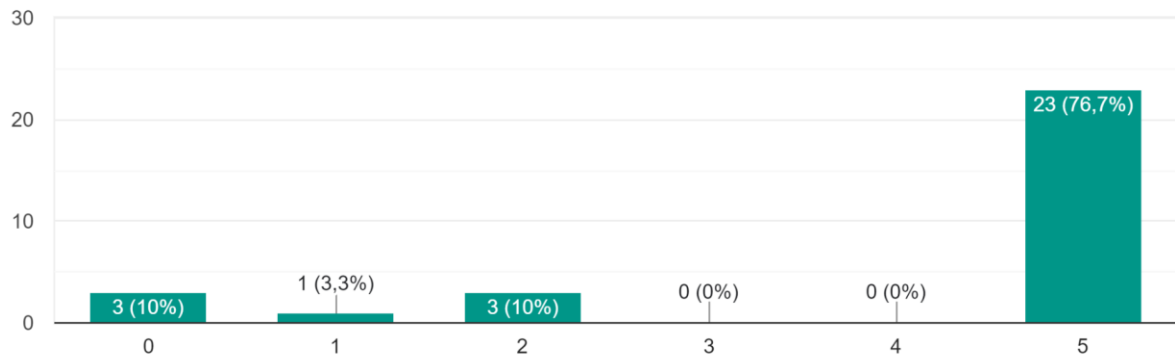
Fonte: elaborado pela autora

Finalmente, a questão 11 verifica a qualidade do caso de uso e a disposição das informações na tela, já que a partir dessas características podemos inferir se um fluxo necessita de uma documentação para ser compreendida ou se apenas a maneira pela qual os dados foram dispostos são suficientes para a compreensão do usuário. Os fluxos com nota 2 ou inferior foram estudados e melhorados (conferir Figura 28).

Figura 28 – Este fluxo foi intuitivo? Respostas

Este fluxo foi intuitivo?

30 respostas



Fonte: elaborado pela autora

Finalizada a sessão de testes, realizamos os ajustes necessários para que o software esteja realizando corretamente os fluxos e, após isso, realizamos a compra de um domínio e assinamos um plano de hospedagem. O domínio foi criado através do site <https://registro.br/> e o plano de hospedagem foi assinado com a plataforma <https://www.pythonanywhere.com/>. Realizado deploy, entramos em contato com alguns voluntários pertencentes ao público-alvo e realizamos os testes de Qualidade de Software.

## 5.2 TESTE DE USABILIDADE E QUALIDADE DE SOFTWARE

Este é o último processo realizado com o software criado. Como explicado pelo Woebcken (2021, n.p.), o objetivo principal deste tipo de teste é: realizar uma busca para “entender como o sistema se comporta no dia a dia, na naturalidade da utilização, e se



ele atende aos requisitos pensados e estabelecidos”. Para aplicar este teste no contexto de uma aplicação de caráter acadêmico, foi decidido seguir o modelo de Teste de aprendizado, em que buscamos compreender o quão fácil é lidar com o software. Seguindo-se a estratégia de criar entrevistas moderadas, foi criado um roteiro de apresentação do uso do aplicativo distinto para cada tipo de perfil e selecionado um grupo de pessoas que representasse o público-alvo em todas as visões do aplicativo: Administrador, Treinador Master, Treinador e Aluno.

No final, tivemos dois testadores da permissão administrador (profissionais da área da Tecnologia), dois testadores da permissão treinador (profissionais da área da Educação Física e Fisioterapia) dois testadores da permissão treinador master (profissionais dos ramos empresarial e gerencial de academias) e dez testadores da permissão aluno (profissionais das TICs).

O teste consiste na execução de uma reunião de cerca de vinte minutos, em que se apresenta o software produzido e, após isso, é passado um questionário avaliando as primeiras impressões do sistema e o interesse de uso.

### **5.2.1 Questionário**

As perguntas foram selecionadas a partir do modelo de questionário de avaliação de usabilidade da empresa JotForm e, das 15 perguntas, foram selecionadas as dez que mais se encaixavam com as necessidades de compreender a usabilidade do software. Reduzimos o número de questões para que o formulário seja o mais atrativo possível para os usuários, tendo em vista a escassez de tempo do público-alvo. As questões – todas em escala de 0 a 5, em que 0 é discordo totalmente e 5 concordo totalmente – selecionadas foram:

1. Acho que gostaria de usar este sistema com frequência.
2. Achei o sistema fácil de usar.
3. As funções deste sistema estavam bem integradas.

4. Imagino que a maioria das pessoas aprenderiam a usar este sistema rapidamente.
5. Achei o sistema muito complicado de usar.
6. Eu preciso aprender um monte de coisas antes de continuar usando este sistema.
7. Eu me senti confortável com este sistema.
8. Foi fácil encontrar a informação de que eu precisava.
9. Eu gostei de usar a interface do sistema.
10. A organização de informações na tela do sistema é clara.

### **5.2.2 Entrevistas**

Durante a fase de entrevistas, buscamos clientes em potencial, incluindo personal studios localizados nos estados de Santa Catarina, Ceará e Maranhão. Antes das entrevistas, realizamos uma pré-filtragem para entender como o nosso produto pode ser útil na rotina de trabalho de cada cliente. Com base nessas informações, criamos um pitch personalizado para cada um deles e o apresentamos nas entrevistas. Recebemos feedbacks positivos dos clientes, o que nos permitiu ajustar o produto para atender melhor suas necessidades.

### **5.2.3 Análise**

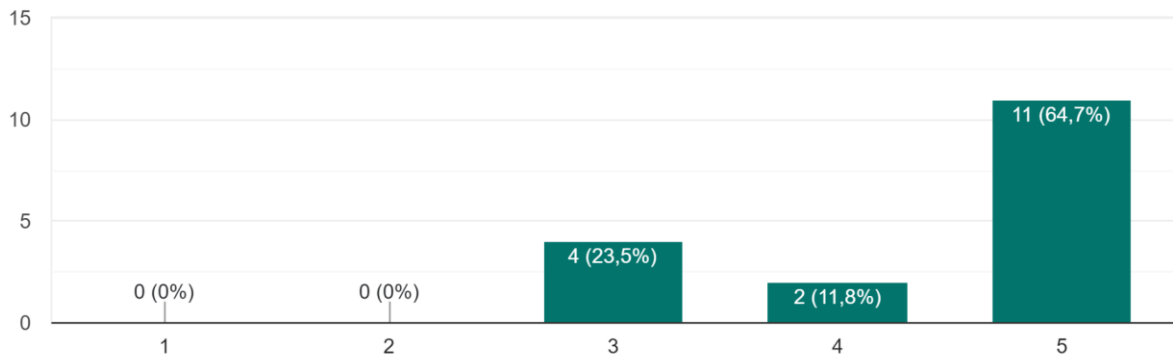
Após 3 semanas de entrevistas, tivemos um total de 20 entrevistas e 17 formulários respondidos. Como algumas respostas tiveram pouca ou nenhuma discrepância, mostraremos aqui apenas os gráficos com mais discrepância entre os resultados.

A primeira questão diz respeito ao interesse na utilização do aplicativo em seu cotidiano, a síntese das respostas está explicitada na Figura 29.

Figura 29 – Resultados questão 01

1. Acho que gostaria de usar este sistema com frequência.

17 respostas



Fonte: elaborado pela autora

Podemos verificar que, mesmo havendo quatro respostas neutras, grande parte do público gostaria de utilizar o software com frequência. Isso nos mostra grande adaptabilidade do sistema ao público-alvo. Outro ponto interessante é que estas quatro respostas neutras vieram das entrevistas focadas na permissão do aluno, que é um usuário opcional do sistema, podendo apenas cadastrar feedbacks e visualizar seus exercícios, sendo compreensível, neste caso, o baixo interesse no sistema.

Com relação à questão 2, 16 pessoas (94,1%) concordaram totalmente com a facilidade no uso, tendo apenas uma (5,9%) pessoa concordado parcialmente. Note-se que esta pergunta diz respeito à permissão treinador master com plus de treinador, a qual não é estudada no escopo inicial. Neste estágio, abrimos apenas as opções das duas permissões, deixando o menu mais extenso e um pouco poluído. Tal conduta torna compreensível a opinião deste usuário, e, mesmo assim, a facilidade do uso pode ser comprovada pela maioria das outras respostas que obtivemos.

Com relação à questão 3, 13 pessoas (76,5%) concordaram totalmente com as funções bem integradas, e quatro pessoas (23,5%) concordaram no que tange a esse ponto. Isso demonstra que o aplicativo foi bem pensado em seu estágio pré-desenvolvimento, comprovando que “A Engenharia de software se ocupa em fornecer as ferramentas necessárias para que programadores e desenvolvedores coloquem a mão na massa, garantindo a unidade e alinhamento do trabalho, gerando sistemas computacionais eficazes e funcionais” (KENZIE, 2021, n.p.).

Com relação à questão 4, três pessoas (17,6%) concordaram e 14 pessoas (82,4%) concordaram totalmente com a crença de que a maioria das pessoas aprenderiam a usar este sistema rapidamente. Este ponto, além de ser outra demonstração de uma engenharia de software bem trabalhada e fundamentada, indica uma boa estruturação da aplicação, tendo como base a interação do usuário, a interface e o *affordance* (características da interface que dão uma ideia de como utilizá-lo). Tal ponto repercute, além disso, segundo os conceitos de Nielsen, um dos critérios mais importantes para a qualidade de um software: usabilidade (resultado da facilidade do uso em conjunto com a satisfação). Esse aspecto também é explorado pela questão 5, da qual quatro pessoas (23,5%) discordaram e 13 pessoas (76,5%) discordaram totalmente da crença de que o sistema é muito complicado de usar e, endossando a qualidade do software.

Com relação à questão 6, sete pessoas (41,2%) discordaram e dez pessoas (58,8%) discordaram totalmente da afirmação de que teriam de iniciar um processo de aprendizado extenso para que utilizasse o sistema. Esta questão visa a identificar a proximidade do sistema com seu público-alvo, de tal modo que, para que haja conforto e uma continuidade de uso, os termos e fluxos devem estar fáceis para os usuários. Com o resultado obtido, podemos identificar uma boa relação entre os conteúdos do sistema e o público.

A questão 7 aborda objetivamente o conforto do usuário usando o sistema, antes mencionado de forma implícita em outras questões. Softwares com essa característica costumam ser utilizados por mais tempo e com maior frequência, algo importante para o

projeto por conta da utilização durante o período de trabalho por parte dos educadores físicos. Nesta questão, notamos que seis pessoas (35,3%) concordaram e 11 pessoas (64,7%) concordam totalmente com o enunciado.

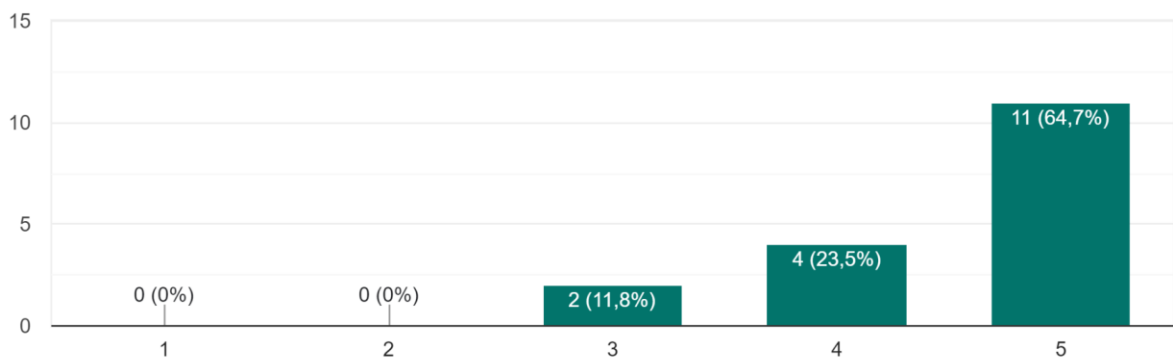
A questão 8 aborda a facilidade em se encontrar as informações necessárias, buscando identificar a qualidade dos fluxos e também investigando se a interface dá destaque às devidas informações. Segundo o grupo de teste, seis pessoas (35,3%) concordaram e 11 pessoas (64,7%) concordaram totalmente. Através desses números, podemos inferir que as informações mais importantes podem ser facilmente localizadas pelo usuário.

A questão 9, embora tenha sido tangenciada indiretamente em outros enunciados da testagem verifica diretamente a aceitação da interface, mensurando com mais precisão: estética, responsividade, facilidade de uso, disposição de informações e objetos interativos. A síntese das respostas está explicitada na Figura 30.

Figura 30 – Resultados questão 09

9. Eu gostei de usar a interface do sistema.

17 respostas



Fonte: elaborado pela autora

Conforme o gráfico indica, uma grande maioria dos usuários teve aceitação total, e, as menores pontuações foram neutras. Assim, podemos verificar que, mesmo com um

bom resultado, pode-se realizar, em melhorias futuras, modificações pequenas (por conta de a estrutura ter agradado em sua maioria) na interface, agregando mais qualidade à interface.

A questão 10 aborda a organização de informações, ponto que avalia se a abstração foi realizada corretamente (tanto em nível de engenharia de software quanto em nível de desenvolvimento), se os fluxos estão congruentes com a disposição das informações e se a interface possui uma estrutura que faça sentido ao uso. Nesta questão, obtivemos 14 pessoas (82,4%) concordando totalmente e três pessoas (17,6%) concordando, revelando bons resultados para este ponto, o qual é de suma importância para realização de decisões estratégicas a partir do sistema.

Este estudo proporcionou resultados promissores que evidenciam uma excelente receptividade e aceitação do software de gerenciamento de máquinas para academias especializadas. Durante as etapas de entrevistas e coleta de formulários, constatou-se que a maioria dos usuários demonstrou grande interesse em utilizar o software de forma frequente. Esse dado ressalta a adaptabilidade notável do sistema ao público-alvo, sinalizando um potencial significativo para sua implementação e adoção no mercado de academias voltadas ao treinamento personalizado.

A receptividade favorável do software reforça a relevância da integração tecnológica no setor de academias e a importância de soluções eficientes de gerenciamento. Ao oferecer um suporte abrangente às atividades dessas academias, o software não apenas simplifica e otimiza os processos de gerenciamento, mas também promove uma experiência aprimorada para os usuários, sejam eles treinadores, alunos ou funcionários.

Esses resultados encorajadores indicam que o software de gerenciamento de máquinas para academias especializadas tem um grande potencial para impulsionar a eficiência operacional e a satisfação dos usuários. Sua capacidade de oferecer uma experiência agradável e intuitiva, juntamente com uma interface amigável e funcionalidades bem integradas, demonstra a atenção cuidadosa dada à usabilidade e ao design centrado no usuário.

No contexto atual, em que a tecnologia desempenha um papel cada vez mais relevante nas atividades diárias, a implementação de soluções inovadoras e adaptadas às necessidades específicas das academias é essencial para se destacar no mercado e proporcionar uma experiência diferenciada aos clientes. O presente estudo fornece uma base sólida e encorajadora para a continuidade desse projeto, com o objetivo de oferecer um software de gerenciamento de máquinas altamente eficaz e adequado às demandas do mercado de academias voltadas ao treinamento personalizado.

### 5.3 SEMELHANÇAS E DIFERENÇAS

Nesta seção, abordaremos as semelhanças e diferenças dos softwares apresentados no capítulo 2.1, intitulado Trabalhos Relacionados. Estes softwares serviram de inspiração para a produção deste trabalho, guiando fluxos e propondo melhorias estruturais para interação com o usuário.

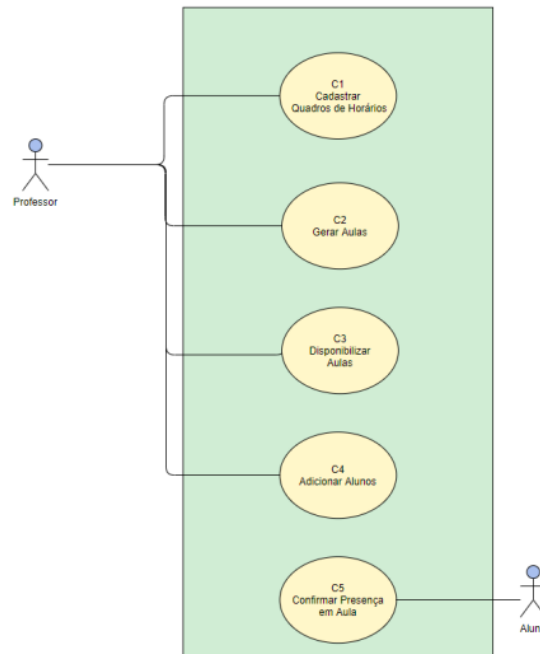
Assim como no capítulo anterior, este é seccionado por software, de modo a facilitar a compreensão.

#### 5.3.1 Requisito de Bacharel

##### *5.3.1.1 Aplicativo mobile para gerenciamento de academias de treinamento funcional*

Analisando o diagrama de Casos de Uso (conferir Figura 31), podem-se visualizar semelhanças entre os softwares, como a predominância de uso por parte do Treinador Físico (que, no trabalho analisado, é chamado de Professor). Além deste aspecto geral, há casos semelhantes, como a Geração de Aulas e o Cadastro de Alunos. Os casos que possuem maior discrepância em relação software proposto neste trabalho não serão aprofundados.

Figura 31 – Diagrama de Casos de Uso



Fonte: Silva (2021, p. 49)

- Geração de Aula

A geração de Aula proposta pelo trabalho analisado neste tópico possui o mesmo objetivo geral da geração de Atendimento: possibilitar a associação de Alunos a Professores em determinado período de tempo e data (verificar Figura 32). Porém, algumas diferenciações são percebidas entre os dois trabalhos, como a configuração do número de Alunos. Assim, no caso da geração de aula, a configuração do número de alunos fica vinculada ao Cadastro do Treinador. O motivo para esta fixação de capacidade máxima por aula tem o objetivo de facilitar a interação do uso por parte do Treinador, de sorte que este indicará o máximo de alunos que consegue atender em um mesmo horário. Ao realizar o preenchimento do formulário de Cadastro de Atendimento, será verificado se o número de Alunos excede o limite, informando-se ao Treinador caso isso ocorra.



Figura 32 – Tela de Geração de Aulas

**Gerar aulas**

Principal

Período

01/09/21 04/09/21

Titulo

Metade da semana pra frente!

Descrição

Não esqueçam de trazer toalha!

Vacancies

4

GERAR

(a) Silva (2021, p. 71)

**Configurar Atendimento**

Segunda	None	None	Editar
Terça	None	None	Editar
Quarta	None	None	Editar
Quinta	None	None	Editar
Sexta	None	None	Editar
Sábado	None	None	Editar
Domingo	None	None	Editar

(b) elaborado pela autora

Define-se, a partir daí, que a geração de Aulas foi tratada de forma diferente na estruturação das aplicações, tornando o fluxo difuso. Neste caso em análise, o fluxo inicia na criação da aula e na liberação para uma quantidade a ser cadastrada de alunos. Assim, o aluno terá de utilizar mais vezes o aplicativo para selecionar suas aulas, o que, embora possa ser apontado como um ponto negativo, confere mais liberdade ao aluno para organizar sua rotina de exercícios.

No software criado neste projeto, é realizado um Cadastro de Horários do Aluno pelo treinador. Após isso, será realizada uma avaliação dos Horários e criados os Atendimentos relacionados à data selecionada no gerenciador de Atendimento. Esta decisão foi tomada durante a coleta de requisitos e a especificação do público-alvo, em que se notou uma necessidade diferente do fluxo especificado neste tópico, tendo em vista a constância na Agenda do Treinador e a regra de negócio número 6, a qual se consubstancia em “Os treinadores realizam um acordo informal de horários de Atendimento com o Aluno e o Cadastra no sistema (RF3)”.

- Cadastro de Alunos

Com relação a este fluxo, as semelhanças são mais aparentes que no tópico anterior (verificar Figura 33), cabendo observar que a criação de um Aluno gera um acesso ao sistema com o menu limitado. O formulário para adição de Alunos da aplicação analisada possui apenas campos obrigatórios: username, informação necessária para a execução do login; nome, informação para identificação por parte do Professor; e senha. Estes dados também são requeridos para a adição de Alunos no software proposto neste trabalho, mas com uma diferença: o username foi trocado pelo campo CPF, apesar do mesmo objetivo. A utilização do CPF em vez de um nome aleatório e único para a realização do login remonta ao conceito definido por Nielsen, em 1993, a respeito de qualidade: "Em geral, as pessoas têm muito mais facilidade em reconhecer algo que lhes é mostrado do que eles ter que recuperar a mesma informação da memória sem ajuda"; e então, para diminuir a necessidade de memorização de um nome aleatório, promovemos a mudança em favor de um valor numérico já conhecido de cada um: o Código de Pessoa Física. Além da facilidade de login, esta escolha trará benefícios para o Treinador na tentativa de criar um Aluno já cadastrado: o software reconhecerá o Cadastro a partir do CPF e entrará em modo de edição, em vez de travar a criação de um novo Aluno por já haver um username idêntico.

Além dos dados obrigatórios, foram incluídas informações extras a respeito do aluno, as quais, apesar de opcionais, servem para delimitação de perfil e facilitam o trabalho do Treinador no momento da instrução, o que é importante também para situações de troca de Treinador.

Figura 33 – Tela de Adição de Aluno

(a) Fonte: Silva (2021, p. 75)

(b) Fonte: elaborado pela autora

### 5.3.1.2 Aplicativo móvel para academia: estudo de tecnologias e desenvolvimento

Analisando as funções criadas no trabalho, podem-se visualizar semelhanças entre os softwares, como o Gerenciamento de Exercícios, a Adição do Exercício a uma lista que será realizada pelo Aluno (chamada de Ficha), a visualização dos Exercícios a serem realizados pelo aluno e a Finalização de séries do Exercício. Estas funções que possuem maior semelhança ao software proposto neste trabalho serão aprofundadas a seguir, organizados por tópicos que levarão o nome da funcionalidade.

- Gerenciamento de Exercícios

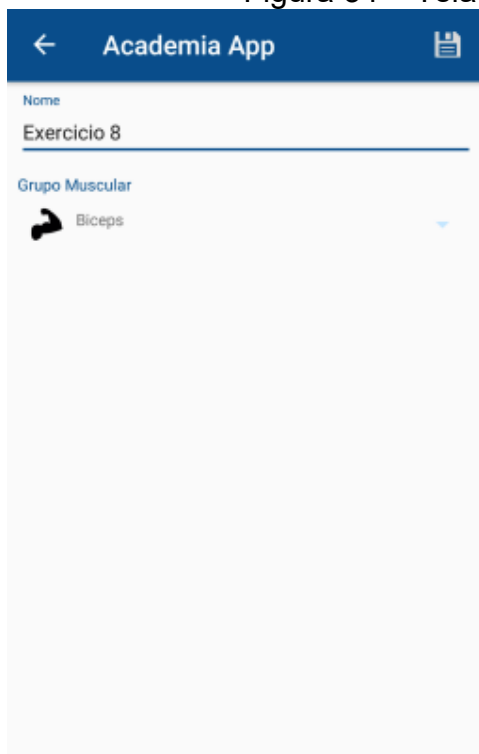
Nesta área, em ambos softwares, é possível realizar todo o CRUD dos Exercícios dentro do Sistema (verificar Figura 34), com o objetivo de tornar mais rápida a associação do Exercício à Ficha (no software proposto neste trabalho, denomina-se Atendimento). Dessa maneira, promove-se um encurtamento do Formulário e se viabiliza a adição de um mesmo Exercício a mais de uma ficha. Neste quesito, este fluxo é semelhante ao software proposto neste trabalho, porém, como o propósito da Ficha é uma repetição do

mesmo grupo de Exercícios e o Atendimento é personalizado em tempo real pelo Treinador, apenas o fluxo de Gerenciamento de Exercício se enquadra como semelhante a este trabalho.

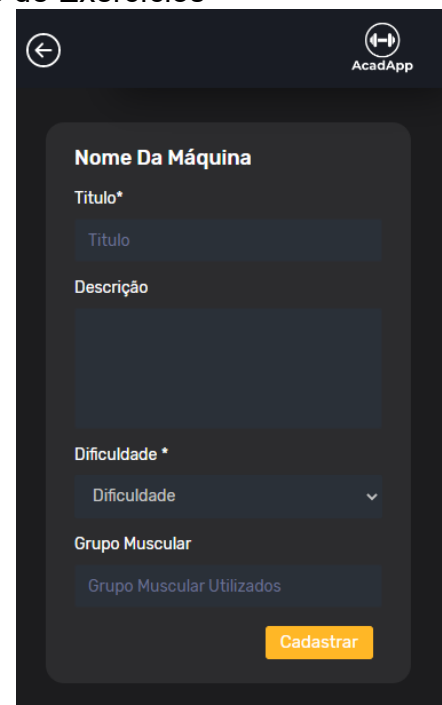
O campo denominado Grupo Muscular possui a finalidade de diferenciar os Exercícios, facilitando a realização de filtragens e abrindo margem para a utilização de algoritmos de inteligência artificial para a recomendação de Exercícios semelhantes, a mesma do software proposto neste trabalho.

Entretanto, é possível observar que o campo utilizado na aplicação estudada é tratado no singular, diferindo do software proposto. Por conta da evolução no campo da Educação Física, vários grupos de músculos podem ser utilizados em um único exercício, então, no software proposto, adaptamos o cadastro de Exercício para permitir a associação entre múltiplos grupos musculares, ou seja, uma adaptação do sistema de tags.

Figura 34 – Tela de Gerenciamento de Exercícios



(a) Fonte: Soares (2016, p. 20)



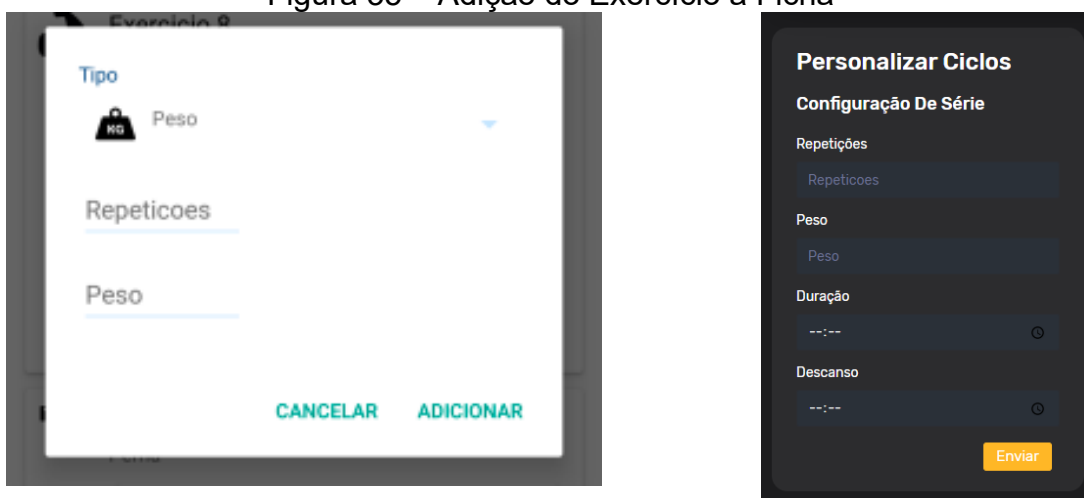
(b) Fonte: elaborado pela autora

- Adição do Exercício à Ficha

Esta área, em ambos softwares, tem por objetivo associar o Exercício com a lista de Exercícios a ser realizado pelo Aluno, identificando características únicas para a realização do Exercício pelo Aluno, como: o número de repetições, o peso a ser utilizado e a quantidade de Séries (verificar Figura 35).

Após análise do aplicativo de mercado Selfit, foram identificados mais dois campos importantes para a aplicação, a saber: o tempo de duração, em casos de Exercícios anaeróbicos; e o tempo de descanso. Ambos os campos foram adicionados na estrutura do software proposto neste trabalho.

Figura 35 – Adição do Exercício à Ficha



(a) Fonte: Soares (2016, p. 20)

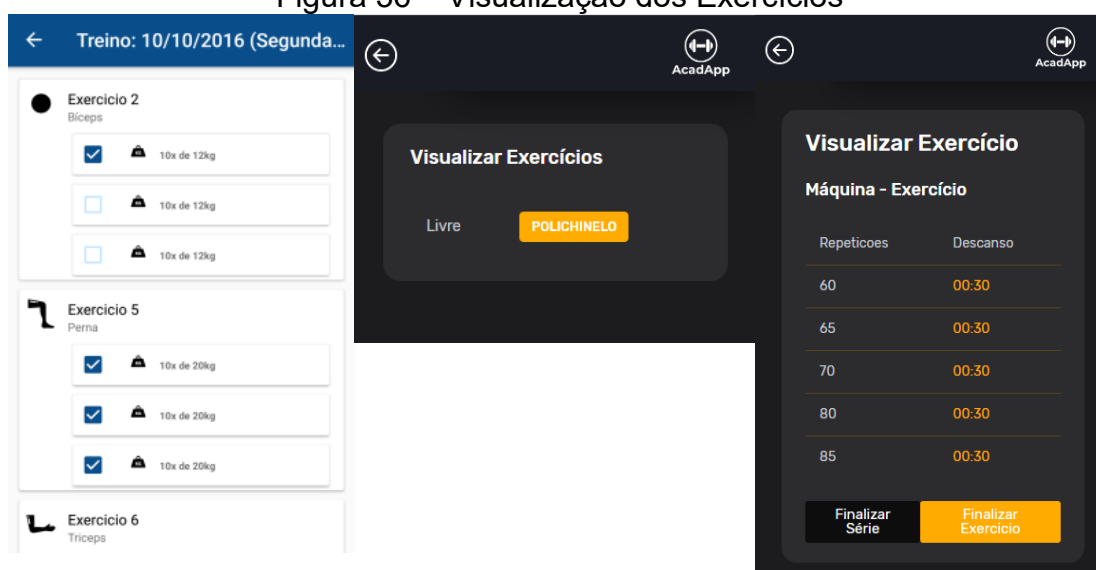
(b) Fonte: elaborado pela autora

- Visualização dos Exercícios e Finalização de séries

Este fluxo do software é voltado ao Aluno, que, através deste, consegue visualizar quais exercícios ele irá realizar e quais já foram realizados (conferir Figura 36). A proposta deste fluxo na aplicação analisada é simples, trazendo, em uma mesma tela, todos os exercícios, de sorte que a finalização da sessão é realizada através de um checkbox, não havendo a possibilidade da finalização do exercício através de um único comando.

No quesito objetivo deste fluxo na aplicação analisada, pode-se dizer que é igual ao fluxo de Visualizar Exercício do software proposto neste trabalho. Entretanto, as diferenças são percebidas ao analisar a estrutura das duas telas. Seguindo a estrutura proposta pela aplicação da empresa Selfit deste fluxo, o software proposto possui duas telas: uma para visualização de todos os exercícios; e outra para visualização dos detalhes e da finalização. Além disso, foi incluído o botão de finalizar exercício, uma opção para a finalização de todas as séries relacionadas àquela atividade. Outra mudança realizada na estrutura foi a escolha de não utilização do checkbox, tendo em vista a possibilidade de o usuário, por erro, desmarcar a sessão. Optamos, então, por transformar a finalização da série em um botão que removerá automaticamente a primeira linha da tabela.

Figura 36 – Visualização dos Exercícios



(a) Fonte: Soares (2016, p. 21)

(b e c) Fonte: elaborado pela autora

## 5.3.2 Mercado

### 5.3.2.1 Aplicativo de treinamento da Selfit

Uma discrepância entre o aplicativo analisado e o que propomos neste trabalho é que este último foi desenvolvido como um software instalável, e, como citado anteriormente, nos objetivos deste trabalho, ele será hospedado em nuvem e acessado via navegador. Os pontos válidos para esta escolha se devem à não necessidade de os usuários instalarem o software, o que contribui por não alocar memória do dispositivo para a utilização da aplicação, além de permitir aos usuários utilizar o software rapidamente e em qualquer dispositivo, ainda que, como sabemos, a maior parte do uso se dê via celulares

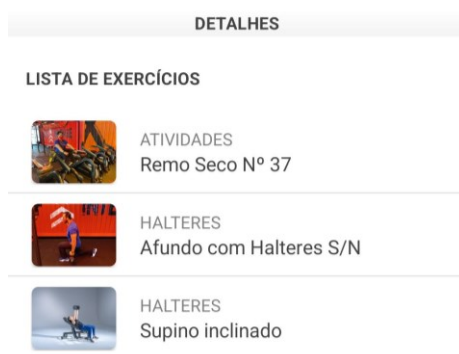
Além deste, o software proposto neste projeto é voltado especificamente ao treinamento personalizado, em contraponto à aplicação estudada neste tópico. Por conta disso, o público-alvo é diferente, e o foco passa a ser facilitar ao profissional em treinamento especializado visualizar máquinas disponíveis para a instrução de exercícios, em contraste com o foco do software apresentado, que busca dar mais autonomia ao usuário.

- **Visualização geral dos exercícios:**

Nesta área, podem-se visualizar todos os exercícios que foram programados pelo treinador e que foram adicionados pelo aluno, podendo-se selecionar um exercício para visualização imediata deste, o que facilita o fluxo para o usuário (conferir Figura 37). Realizada a análise das informações retornadas nesta área da página, pode-se concluir que o posicionamento das informações é de fácil aprendizagem, o que é um aspecto positivo para a aplicação, tendo em vista o alto fluxo de alunos na empresa.

Na lista de exercícios, são retornadas informações que facilitam a compreensão do exercício a ser realizado: máquina a ser utilizada, nome do exercício e imagem da atividade. Trazendo as informações pertinentes de forma simples e agradável à interação do usuário, a estrutura serviu de modelo para o desenvolvimento do software a ser produzido.

Figura 37 – Área inferior da página de Visualização Geral de Exercícios - Listagem de Exercícios



(a) Fonte: Selfit (s.d)

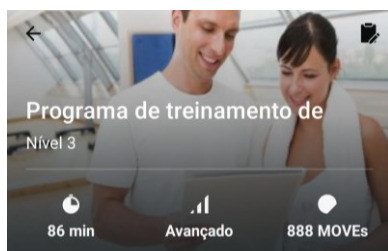


(b) Fonte: elaborado pela autora

Na parte superior da tela, há informações a respeito do treinamento do dia: tempo total, nível e qual a pontuação para a realização completa deste, uma espécie de *gameficação* realizada pela empresa (verificar Figura 38). Logo abaixo, é possível acessar a área de emparelhamento do aplicativo à cinta de frequência cardíaca e, caso haja o emparelhamento anterior, este será informado no mesmo local. Essas informações são úteis para a realização do monitoramento do treino por parte do aluno, levando em conta que o treino pode ser alterado pelo discente, selecionando o item superior direito. Em contrapartida, esta área não tem serventia ao software a ser desenvolvido, já que o público-alvo são empresas que realizam apenas o treinamento personalizado, ou seja, os alunos não terão autonomia para a alteração do treinamento. Assim, por conta da análise da área identificando sua não utilização, ela será removida. Já o posicionamento do botão de Voltar foi identificado como estratégico, pois outros aplicativos, como Instagram e WhatsApp, também o utilizam, facilitando a aprendizagem do uso (conferir Figura 39). Logo, inserimos à estrutura a localização fixa do botão Voltar na mesma área.



Figura 38 – Área superior da página de Visualização Geral dos Exercícios - Detalhes do Treinamento



CINTA DE FREQUÊNCIA CARDÍACA

Cinta de frequência cardíaca não emparelhada  
Clique para emparelhar uma cinta de frequência cardíaca

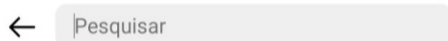


(a) Fonte: Selffit (s.d)

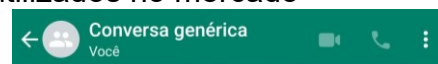


(b) Fonte: elaborado pela autora

Figura 39 – Área superior de softwares utilizados no mercado



(a) Fonte: aplicativo Instagram (s.d)



(b) Fonte: aplicativo WhatsApp (s.d)

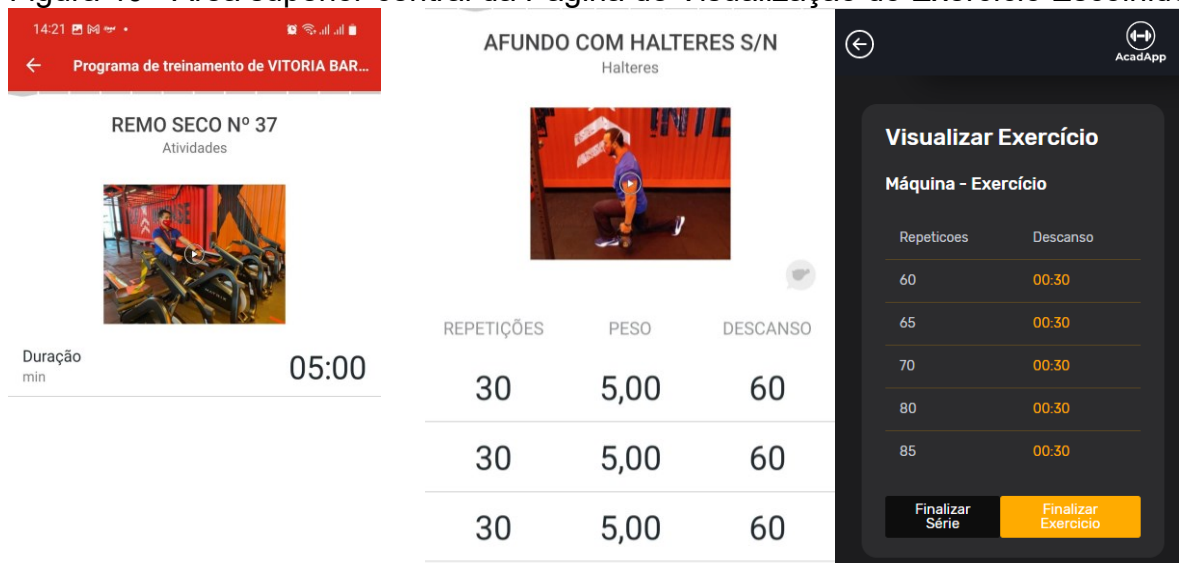
(a) retirado da página de pesquisa

(b) retirado da página de mensagens

- Visualização do Exercício Escolhido:

Nesta página, podem-se visualizar com maiores detalhes informações sobre o exercício selecionado, como o tempo de duração ou a quantidade de repetições, o peso que deve ser utilizado e o tempo de descanso, além de um vídeo demonstrativo do exercício (conferir Figura 40). Como o aplicativo é voltado para o uso do aluno, o tipo de atividade retorna uma tabela diferente, simplificando a visualização. Porém, por conta do cronograma do projeto e da não alteração da qualidade do produto, seguimos a mesma estrutura geral da visualização do exercício, com a simplificação da página para retornar a mesma tabela, alterando, apenas, o título das colunas.

Figura 40 - Área superior-central da Página de Visualização do Exercício Escolhido



(a) e (b) Fonte: aplicativo Selfit (s.d)

(c) Fonte: elaborado pela autora

(a) exercício aeróbico

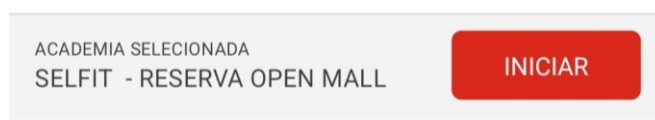
(b) exercício anaeróbico

(c) exercício

(Imagem do protótipo da descrição de exercício, pertencente à página de Visualização de Exercício, retirado do Figma, capturado pela autora)

Com relação à parte inferior da página, caso o treinamento não tenha sido iniciado, na página de Visualização Geral de Exercícios será apresentado um botão chamado Iniciar (conferir Figura 41). Este fluxo foi removido da construção do Software, tendo em vista que o início do Atendimento será feito através do horário cadastrado pelo Treinador. Outro ponto removido foram as informações da Academia, pois, sendo está uma rede unificada e com várias Academias pertencentes à empresa, é importante que o Aluno tenha ciência da Academia em que ele iniciará o treinamento. Em nosso caso, porém, isso se torna desnecessário, já que nosso público-alvo são academias únicas e independentes.

Figura 41 - Área inferior da Página de Visualização do Exercício Escolhido antes de iniciar treinamento



(a) Fonte: Aplicativo Selfit (s.d)

Quando iniciado o treinamento, a parte inferior é trocada para o botão de finalizar o Exercício, indicando que este já foi realizado pelo aluno (conferir Figura 42). Esta funcionalidade mostrou-se relevante para o software a ser desenvolvido, pois diminui o uso da Finalização do Exercício por parte do Treinador, reduzindo um fluxo. Acrescentamos o botão de Finalizar Série, para que o Aluno tenha o controle de quantas repetições da série ele terá de fazer. Porém, quando faltar apenas uma repetição, aparecerá apenas o botão de Finalizar Exercício.

Figura 42 – Área inferior da Página de Visualização do Exercício Escolhido após iniciar treinamento



(a) Fonte: Aplicativo Selfit (s.d)

(b) Fonte: elaborado pela autora

### 5.3.3 Tabela de Semelhanças e Diferenças

Neste tópico, será realizada uma análise comparativa entre os diferentes softwares citados no capítulo 2.1, intitulado Trabalhos Relacionados. O objetivo é avaliar as funcionalidades oferecidas por cada um deles e identificar suas semelhanças e diferenças. Essa análise será apresentada na forma de uma tabela (conferir Tabela 5), onde as funcionalidades são listadas e marcadas com um "✓" para indicar a presença em cada software avaliado.

A tabela de semelhanças e diferenças apresentada neste capítulo permitirá uma visão clara e comparativa das funcionalidades oferecidas por cada software analisado. Essa análise será um recurso valioso para auxiliar na escolha do software mais adequado às necessidades da academia, considerando suas particularidades e objetivos.

Tabela 5 – Semelhanças e diferenças

<b>Funcionalidade</b>	<b>Selfit</b>	<b>Silva</b>	<b>Soares</b>	<b>AcadApp</b>
<b>Geração de Aulas</b>	✓	✓	✓	✓
<b>Acompanhamento de Medidas</b>	✓		✓	✓
<b>Permissionamento de Usuário</b>	✓	✓		✓
<b>Geração de Treinamento Personalizado</b>				✓
<b>Feedback do Treino</b>	✓			✓
<b>Geração de Ficha</b>	✓		✓	
<b>Confirmação de Presença</b>		✓		
<b>Progresso do Treino</b>	✓	✓		✓
<b>Gameificação</b>	✓			
<b>Disponibilização de Aulas</b>		✓		
<b>Gestão de Aparelhos em Tempo Real</b>				✓
<b>Validação do Máximo de Alunos</b>	✓	✓		✓
<b>Associar Grupo Muscular ao Exercício</b>	✓		✓	✓

Fonte: elaborado pela autora

## 6 CONCLUSÃO

Ao longo deste projeto de desenvolvimento do AcadApp, um Software de Gestão de Máquinas para Academia de Atividade Física, pudemos explorar as diversas oportunidades de integração tecnológica no mercado fitness, que, como sabemos, está em constante crescimento. A popularização dos smartphones possibilitou o acesso rápido e prático a informações por meio de aplicativos móveis dedicados.

Uma das principais vantagens encontradas foi a possibilidade de disponibilizar o sistema na nuvem, aproveitando a infraestrutura de rede existente e eliminando a necessidade de ocupar a memória dos dispositivos. Além disso, o uso do framework Django agilizou o desenvolvimento, reduzindo a repetição de código e oferecendo recursos de segurança, gerenciamento de usuários e controle de permissões.

A aplicação dos processos descritos na engenharia de software foi fundamental para garantir a qualidade do software e facilitar a comunicação entre os diferentes participantes do projeto. A escolha do framework Scrum para gerenciar o processo de desenvolvimento mostrou-se adequada, considerando as características específicas do projeto, como a disponibilidade semanal da orientadora e a realização das cerimônias do Scrum.

É importante ressaltar que a engenharia aplicada na estruturação do software teve que acompanhar o processo criativo, as mudanças no mercado e os problemas inesperados que surgiram durante o desenvolvimento. A capacidade de adaptação e flexibilidade foram essenciais para superar os desafios encontrados ao longo do projeto.

Em suma, este projeto demonstrou a importância da integração tecnológica no mercado fitness, bem como a aplicação adequada de processos de engenharia de software. Acreditamos que o software de gestão de aparelhos de academia desenvolvido atende às necessidades do mercado e contribuirá para otimizar os processos de gerenciamento nesse setor em constante evolução.

O objetivo principal da aplicação foi promover uma comunicação visual, intuitiva e simples para identificação de aparelhos livres, assim como facilitar o gerenciamento de atendimentos. Essa meta foi alcançada com sucesso, pois para os treinadores conseguirem encaminhar seus alunos aos aparelhos, estes precisam estar livres e o gerenciamento de atendimento unificado para todos os treinadores de uma mesma academia facilita o controle e a organização. Além disso, a possibilidade de cadastrar alunos apenas em horários que não ultrapassem o máximo de alunos permitidos por horário evita sobrecarga e assegura um ambiente adequado para todos os usuários do sistema. Com a implementação dessas funcionalidades, o AcadApp busca oferecer uma experiência aprimorada tanto para os gestores das academias quanto para os treinadores e alunos, promovendo uma gestão eficiente e proporcionando uma experiência satisfatória durante o uso do software.

## 6.1 MELHORIAS FUTURAS

Nesta sessão, serão abordados alguns tópicos referentes ao que foi observado nas etapas de desenvolvimento e testes como possíveis melhorias a serem implementadas. Porém, como tais aspectos não haviam sido estudados e inseridos no escopo do projeto, eles não foram desenvolvidos, mas ficarão aqui registrados. Este tópico também serve de convite para os alunos do curso que têm interesse em trabalhar com sistemas web e gostariam de aprofundar seus conhecimentos utilizando este projeto e as seguintes propostas de melhorias. Separaremos o conteúdo em tópicos para facilitar a compreensão.

### 6.1.1 Treinador Master com acesso a Fluxos de Treinador

Quando criado o documento de especificação de requisitos, o usuário com Permissão de Treinador Master seria uma espécie de gestor, isto é, um agente dentro da academia que seria responsável pela gestão de itens mais amplos dentro da academia.

Podemos imaginar o usuário desta permissão como o dono da academia, um secretário ou até mesmo um agente do sistema que se dispõe a realizar os cadastros iniciais para utilização do sistema. O que não foi previsto é que, em alguns desses casos, o dono da academia e treinador são a mesma pessoa, o que o sistema não estava preparado para comportar.

Atualmente, no cadastro do Treinador Master, existe uma opção de Treinador, a qual amplia as opções do Menu para as opções das duas permissões. Esta foi uma solução rápida para que as entrevistas pudessem seguir normalmente, pois, além de criar um exagero de informações, cria fluxos desnecessários, diminuindo a qualidade do produto.

### **6.1.2 Plano de Aulas**

Como o produto gerado foi específico para estúdios de treinamento personalizados, o encaminhamento de atividades aos alunos pode ser feito apenas em tempo real, de acordo com as máquinas que se encontram livres para utilização. Porém, para diminuir o tempo de utilização do treinador no sistema, pode-se implementar a ideia de um plano de atividades e, em conjunto, um plano de aulas. Para a implementação de um plano de atividades, o controle do uso dos aparelhos seria feito na visualização das atividades pendentes, em vez de isso ocorrer no momento do encaminhamento. Já para a implementação do plano de aulas, seria necessário ter um controle da aula que será iniciada e do prazo para a atualização deste plano.

### **6.1.3 Permitir ao Aluno realizar a edição do Treinamento**

Uma melhoria do software que pode ser realizada após a implementação do Plano de Atividades é a edição deste pela permissão do aluno. Esta otimização permite ao aplicativo atender a um público diferente daquele inicialmente previsto, ou seja,

academias sem treinamento personalizado, em que onde é criado um Plano de Aulas, cabendo ao aluno executá-lo ou modificá-lo.

Durante a edição do treinamento, o aluno teria permissão para alterar o número de séries, a configuração de cada série, além de adicionar ou excluir exercícios.

Para que esta modificação seja realizada, seria necessário realizar modificações na estrutura da tela de visualizar exercícios e na tela de visualizar exercício.

#### **6.1.4 Importação**

Antes da utilização do aplicativo no cotidiano do profissional da Saúde, alguns cadastros iniciais devem ser realizados, tais como: cadastro de máquinas, exercícios e alunos. Por conta de a implantação do software ser realizada após a inicialização da academia, torna-se exaustiva a repetição dos fluxos deste cadastro. Este tópico do trabalho gira em torno desta questão.

A importação consiste na geração de um documento, normalmente uma planilha, específico para cada objeto e com informações dos cadastros. Este documento será enviado para o sistema via upload, de tal modo que as linhas serão analisadas e gerarão cadastros com as informações contidas em cada célula, possibilitando, assim, o cadastro de diversos itens em um único fluxo.

#### **6.1.5 Integração com outras plataformas**

Com as bigtechs dominando o mercado tecnológico, as integrações com as mais conhecidas tornam o aplicativo mais aceitável e dinâmico. Assim, integrações com empresas como Google ou Facebook são bastante comuns e trazem valor à aplicação.

Algumas integrações possíveis nesta aplicação seriam a utilização da conta Google/Facebook para realizar login: neste caso, seria apenas necessário que o usuário com uma das permissões acima cadastrasse seu e-mail relacionado a uma das contas. Algumas bibliotecas do Django já fazem esta integração, de maneira que o usuário



apenas teria de realizar algumas modificações nas validações para que fosse possível encontrar a permissão do usuário a partir da conta integrada.

Outra integração interessante para o software é a criação de Eventos no Google Calendar do aluno, associando a ele seu Atendimento. Assim, é possível utilizar o sistema de notificação do Calendar em múltiplos aparelhos, demandando um tempo de implementação menor do que caso fosse implementado do zero.

#### **6.1.6 Permitir ao Aluno realizar a Troca de Horário**

Esta modificação vai um pouco contra ao escopo inicial do projeto, porém, a funcionalidade de alteração ou realização de pedido para alteração de Horário de um Atendimento específico traz autonomia para o Aluno e diminui a necessidade de utilização de outros aplicativos, como o WhatsApp, para a gestão dos Atendimentos.

Para essa melhoria, seria necessário criar uma nova página com suas pesquisas no banco de dados e validações. Uma das validações mais importantes seria a respeito da lotação de horário, para que não haja mais alunos do que se pode atender.

## REFERÊNCIAS

ALÉSSIO, Simone; SABADIN, Neli; ZANCHETT, Pedro. **Processos de Software**. Itaipava: UNIASSELVI, 2017.

AUDY, Jorge. **Scrum 360**: Um guia completo e prático de agilidade. São Paulo: Casa do Código, 2015.

BORDIN, Andréa. **Análise e Especificação de Requisitos**: Engenharia de Software I. Araranguá: Moodle, 2021. Disponível em: <https://moodle.ufsc.br/mod/resource/view.php?id=2815591>. Acesso em: 05 jul. 2022.

BORDIN, Andréa. **Análise e Especificação de Requisitos**: Modelagem Conceitual Estrutural. Araranguá: Moodle, 2021. Disponível em: <https://moodle.ufsc.br/mod/resource/view.php?id=2846292>. Acesso em: 05 jul. 2022.

BORDIN, Andréa. **Requisitos de Software**: Engenharia de Software I. Araranguá: Moodle, 2021. Disponível em: <https://moodle.ufsc.br/mod/resource/view.php?id=2789972>. Acesso em: 04 jul. 2022.

CHUNG, Lawrence et al. **Non-Functional Requirements in Software Engineering**. 2000. ed. New York: Springer Science + Business Media, LLC, 1999.

COUTINHO, Thiago. O que é Sprint Scrum? Aprenda seu conceito e como ele se aplica na prática!: Veja como funciona a Sprint Scrum, quais são as suas características e entenda por que é chamada de coração dessa metodologia ágil. **Voitto**. [S.l.], 31 out. 2018. Disponível em: <https://www.voitto.com.br/blog/artigo/sprint-scrum>. Acesso em: 06 jul. 2022.

DOMÍNGUEZ, Arturo. Engenharia de Software. **StuDocu**. Alagoas, 30 ago. 2010. Disponível em: <https://www.studocu.com/pt-br/document/universidade-estacio-de-sa/processos-de-desenvolvimento-de-software/material-didatico-engenharia-de-software/12932933>. Acesso em: 04 jul. 2022.

IBM. Rational Software Architect Standard Edition / 7.5.0: Diagramas de Classes. **IBM Corporation**. [S.l.], 05 mar. 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>. Acesso em: 19 jun. 2022.

JotForm. **Questionário de avaliação de usabilidade**. Disponível em: <https://www.jotform.com/pt/form-templates/questionario-de-avaliacao-de-usabilidade>. Acesso em: 25 abr. 2023.

KAMADA, Aqueo. **Execução de Serviços Baseada em Regras de Negócio**. 2006, 222 f. Tese de Doutorado (Engenharia Elétrica) - Universidade Estadual De Campinas, São Paulo, Campinas, SP, 2006. Disponível em: <https://repositorio.unicamp.br/Busca/Download?codigoArquivo=467308>. Acesso em 05 jun. 2023.

**KENZIE**. [S./], 15 set. 2021. Disponível em: <https://kenzie.com.br/blog/engenharia-de-software-2/>. Acesso em: 05 jul. 2022.

LARMAN, Craig. **Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development**. 3. ed. Saddle River: Pearson Education, 2004.

LERCHE-JENSEN, Steen. Criando e estimando tarefas. **Scrum.as Academy**. [S./], 24 mai. 2019. Disponível em: <https://www.scrum.as/academy.php?show=6&chapter=11&name=%20Criando%20e%20Oestimando%20tarefas>. Acesso em: 19 jun. 2022.

MARTINS, José. **Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML**. 4. ed. Rio de Janeiro: Brasport, 2007.

NIELSEN, Jakob. **Usability Engineering**. Cambridge: Academic Press Inc, 1993.

NOCRATO, Diane. Unifor. **Mercado de trabalho: cresce a atuação do “personal trainer”**: O mercado de orientação e supervisão de atividade física está em ascensão para os profissionais de Educação Física. Unifor. Cidade, data no formato 15 dez. 2020. Disponível em: <https://unifor.br/web/graduacao/-/mercado-de-trabalho-cresce-a-atuacao-do-personal-trainer-#:~:text=Mesmo%20em%20momentos%20de%20crise,da%20pr%C3%A1tica%20da%20atividade%20f%C3%ADsica>. Acesso em: 31 maio 2022.

PRESSMAN, Roger. **Engenharia de Software**. 3. ed. São Paulo: Markron Books, 1995.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software: Uma Abordagem Profissional**. 9. ed. Porto Alegre: AMGH, 2021.

SABBAGH, Rafael. **Scrum: Gestão Ágil para Projetos de Sucesso**. São Paulo: Casa do Código, 2013.

SANTOS, Aline; RADAEL, Edilson; PINHATA, Rosimeire; ROSSI, Sandreliz; MENDES, Sandro. Diretrizes para a Construção de Casos de Uso Eficazes. **Revista de Informática Aplicada**, Santo André, v. 2, n. 2, p. 70-75 dez. 2006.

Selfit. Conheça a Selfit: Nosso Manifesto. **Selfit**. [S.l.], [s.d.]. Disponível em: <https://old.selfitacademias.com.br/conheca-a-selfit/>. Acesso em: 20 set. 2022.

SILVA, Geison Machado da. **Aplicativo Mobile para gerenciamento de academias de treinamento funcional**. 2021. 96 f. Trabalho de Conclusão de Curso (Sistemas de Informação) – Universidade Federal de Santa Catarina, Florianópolis, SC, 2021. Disponível em: [https://repositorio.animaeducacao.com.br/bitstream/ANIMA/19940/1/GEISON\\_MACHADO\\_DA\\_SILVA-%5B73218-904-1-980558%5DTCC.pdf](https://repositorio.animaeducacao.com.br/bitstream/ANIMA/19940/1/GEISON_MACHADO_DA_SILVA-%5B73218-904-1-980558%5DTCC.pdf). Acesso em: 10 jun. 2023.

SILVA, Jean; OLIVEIRA, Luciano. O Personal Trainer e sua Qualificação Profissional. **Revista Campo do Saber**, Cabedelo, v. 3, n. 2, p. 114-128, dez. 2017.

SOARES, Marco Aurélio. **Aplicativo móvel para academia: estudo de tecnologias e desenvolvimento**. 2016. 69 f. Trabalho de Conclusão de Curso (Sistemas de Informação) – Instituto Federal de Minas Gerais, Sabará, 2016. Disponível em: <https://www.ifmg.edu.br/sabara/biblioteca/trabalhos-de-conclusao-de-curso/tcc-documentos/TCCMarcoAurelio.pdf>. Acesso em: 10 jun. 2023.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Prentice Hall, 2011.

THOMPSON, Walter. Worldwide Survey of Fitness Trends for 2022. **ACSM's Health & Fitness Journal**, Indianapolis, v. 26, n. 1, p. 11-20, jan. 2022. Disponível em: [https://journals.lww.com/acsm-healthfitness/fulltext/2022/01000/worldwide\\_survey\\_of\\_fitness\\_trends\\_for\\_2022.6.aspx](https://journals.lww.com/acsm-healthfitness/fulltext/2022/01000/worldwide_survey_of_fitness_trends_for_2022.6.aspx). Acesso em: 31, mai. e 2022.

TURINE, Marcelo Augusto Santos; MASIERO, Paulo Cesar. **Especificação de requisitos: uma introdução**. São Carlos: Instituto de Ciências Matemáticas de São Carlos, Universidade de São Paulo, 1996. Disponível em: [http://www2.unemat.br/rhycardo/download/engenharia\\_de\\_requisitos.pdf](http://www2.unemat.br/rhycardo/download/engenharia_de_requisitos.pdf). Acesso em: 25 abr. 2023.

VIANNA, Ralph. **A tecnologia e o empreendedorismo aliados ao Personal Trainer**. IESPE. Juiz de Fora, 2 ago. 2022. Disponível em: <https://www.iespe.com.br/blog/a-tecnologia-e-o-empreendedorismo-aliados-ao-personal-trainer/>. Acesso em: 28 jun. 2022.

WIEGERS, Karl; BEATTY, Joy. **Software Requirements**. 3. ed. Redmond: Microsoft Press, 2013.

WOEBCKEN, Cayo. **Rockcontent**. [S.l.], 26 maio 2021. Disponível em: <https://rockcontent.com/br/blog/teste-de-usabilidade/>. Acesso em: 05 jul. 2022.

ZANCHETT, Pedro. **Engenharia e Projeto de Software**. Indaial: UNIASSELVI, 2015.

## TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

### IV - CONSENTIMENTO LIVRE E ESCLARECIDO - Resolução 196/96 CONEP

O respeito devido à dignidade humana exige que toda pesquisa se processe após consentimento livre e esclarecido dos sujeitos, indivíduos ou grupos que por si e/ou por seus representantes legais manifestem a sua anuência à participação na pesquisa.

IV.1 - Exige-se que o esclarecimento dos sujeitos se faça em linguagem acessível e que inclua necessariamente os seguintes aspectos:

- a) a justificativa, os objetivos e os procedimentos que serão utilizados na pesquisa;
- b) os desconfortos e riscos possíveis e os benefícios esperados;
- c) os métodos alternativos existentes;
- d) a forma de acompanhamento e assistência, assim como seus responsáveis;
- e) a garantia de esclarecimento, antes e durante o curso da pesquisa, sobre a metodologia, informando a possibilidade de inclusão em grupo controle ou placebo;
- f) a liberdade do sujeito de recusar a participar ou retirar seu consentimento, em qualquer fase da pesquisa, sem penalização alguma e sem prejuízo ao seu cuidado;
- g) a garantia do sigilo que assegure a privacidade dos sujeitos quanto aos dados confidenciais envolvidos na pesquisa;
- h) as formas de ressarcimento das despesas decorrentes da participação na pesquisa; e
- i) as formas de indenização diante de eventuais danos decorrentes da pesquisa.

IV.2 - O termo de consentimento livre e esclarecido obedecerá aos seguintes requisitos:

- a) ser elaborado pelo pesquisador responsável, expressando o cumprimento de cada uma das exigências acima;
- b) ser aprovado pelo Comitê de Ética em Pesquisa que referenda a investigação;
- c) ser assinado ou identificado por impressão dactiloscópica, por todos e cada um dos sujeitos da pesquisa ou por seus representantes legais; e
- d) ser elaborado em duas vias, sendo uma retida pelo sujeito da pesquisa ou por seu representante legal e uma arquivada pelo pesquisador.

IV.3 - Nos casos em que haja qualquer restrição à liberdade ou ao esclarecimento necessários para o adequado consentimento, deve-se ainda observar:

- a) em pesquisas envolvendo [crianças e adolescentes](#), portadores de perturbação ou doença mental e sujeitos em situação de substancial diminuição em suas capacidades de consentimento, deverá haver justificativa clara da escolha dos sujeitos da pesquisa, especificada no protocolo, aprovada pelo Comitê de Ética em Pesquisa, e cumprir as exigências do consentimento livre e esclarecido, através dos representantes legais dos referidos sujeitos, sem suspensão do direito de informação do indivíduo, no limite de sua capacidade;
- b) a liberdade do consentimento deverá ser particularmente garantida para aqueles sujeitos que, embora adultos e capazes, estejam expostos a condicionamentos específicos ou à influência de autoridade, especialmente estudantes, militares, empregados, presidiários, internos em centros de readaptação, casas-abrigo, asilos, associações religiosas e semelhantes, assegurando-lhes a inteira liberdade de participar ou não da pesquisa, sem quaisquer represálias;
- c) nos casos em que seja impossível registrar o consentimento livre e esclarecido, tal fato deve ser devidamente documentado com explicação das causas da impossibilidade e parecer do Comitê de Ética

em Pesquisa;

d) as pesquisas em pessoas com o [diagnóstico de morte encefálica](#) só podem ser realizadas desde que estejam preenchidas as seguintes condições:

- documento comprobatório da morte encefálica (atestado de óbito);
- consentimento explícito dos familiares e/ou do responsável legal, ou manifestação prévia da vontade da pessoa;
- respeito total à dignidade do ser humano sem mutilação ou violação do corpo;
- sem ônus econômico financeiro adicional à família;
- sem prejuízo para outros pacientes aguardando internação ou tratamento;
- possibilidade de obter conhecimento científico relevante, novo e que não possa ser obtido de outra maneira;

e) em comunidades culturalmente diferenciadas, inclusive [indígenas](#), deve-se contar com a anuência antecipada da comunidade através dos seus próprios líderes, não se dispensando, porém, esforços no sentido de obtenção do consentimento individual;

f) quando o mérito da pesquisa depender de alguma restrição de informações aos sujeitos, tal fato deve ser devidamente explicitado e justificado pelo pesquisador e submetido ao Comitê de Ética em Pesquisa. Os dados obtidos a partir dos sujeitos da pesquisa não poderão ser usados para outros fins que os não previstos no protocolo e/ou no consentimento.