



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
BACHAREL EM SISTEMAS DE INFORMAÇÃO

APLICATIVO MÓVEL PARA ACADEMIA: ESTUDO DE TECNOLOGIAS E DESENVOLVIMENTO

MARCO AURÉLIO SOARES

Orientador: Cristiano Santos Botelho
Instituto Federal de Minas Gerais - Campus Sabará

SABARÁ
DEZEMBRO DE 2016

MARCO AURÉLIO SOARES

**APLICATIVO MÓVEL PARA ACADEMIA: ESTUDO DE
TECNOLOGIAS E DESENVOLVIMENTO**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Sabará, como requisito fundamental para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Cristiano Santos Botelho
Instituto Federal de Minas Gerais - Campus Sabará

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
BACHAREL EM SISTEMAS DE INFORMAÇÃO
SABARÁ
DEZEMBRO DE 2016

Biblioteca IFMG *Campus* Sabará

S676a Soares, Marco Aurélio
Aplicativo móvel para academia [manuscrito] : estudo de tecnologias e desenvolvimento. / Marco Aurélio Soares. - 2016.

70 f.: il.

Orientador: Prof. Cristiano Santos Botelho

Monografia (Bacharelado em Sistemas de Informação) – Instituto Federal de Minas Gerais, *Campus* Sabará.

1. Aplicativos móveis. – Monografia. 2. Android (Recurso eletrônico). – Monografia. 3. Software - Desenvolvimento. – Monografia. I. Botelho, Cristiano Santos. II. Instituto Federal de Minas Gerais, *Campus* Sabará, Bacharelado em Sistemas de Informação. III. Título.

CDU 004.451.9

ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO de MARCO AURÉLIO SOARES

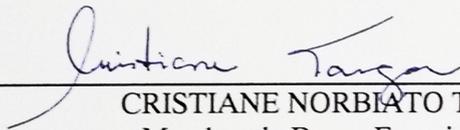
No dia 06 do mês Dezembro do ano de 2016, às 11:40 horas, os professores: Cristiane Norbiato Targa, Cristiano Santos Botelho e João Eduardo Montandon compareceram para defesa pública do Trabalho de Conclusão de Curso intitulado APLICATIVO MÓVEL PARA ACADEMIA: ESTUDO DE TECNOLOGIAS E DESENVOLVIMENTO, requisito obrigatório para a obtenção do título de Bacharel em Sistemas de Informação. Após a apresentação e as observações dos membros da banca avaliadora, ficou definido que o trabalho foi considerado:

(x) Aprovado () Reprovado.

O resultado final foi comunicado publicamente ao candidato pelo Professor Orientador. Nada mais havendo a tratar, o Professor Orientador a reunião e lavrou a presente ATA, que será assinada por todos os membros participantes da banca avaliadora.



CRISTIANO SANTOS BÔTELHO
Professor Orientador



CRISTIANE NORBIATO TARGA
Membro da Banca Examinadora



JOÃO EDUARDO MONTANDON
Membro da Banca Examinadora



MARCO AURÉLIO SOARES
Aluno(a)

Resumo

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.

Palavras-chave: Dispositivos Móveis, Aplicativos. Academia. Android. Desenvolvimento de *software*.

Abstract

Information technologies are becoming more and more essential in people's lives, and with the popularization of mobile devices in recent years, there is a demand for applications that facilitate the day-to-day tasks of those using the technology. At the gyms, there is a need to manage the workout sheets of their students and a physical accompaniment through physical evaluations. However, most of the time, all of this management is done manually, which generates enormous manual work, becoming a tiresome task for teachers, as well as generating paper expenses. The objective of this work was to study the latest technologies in the area of mobile applications, and with that develop an application containing functions of management of workout sheets, training and physical evaluations. At the end of this work, it was possible to obtain a functional and viable product, containing functionalities to aid in the activities of the gym, besides containing differential characteristics such as offline operation and availability of data.

Keywords: Mobile Devices, Applications. Gym. Android. Software Development.

Lista de Figuras

Figura 1 – Exemplos de telas do sistema operacional Android, versão Lollipop (5.0)	6
Figura 2 – Gráfico comparativo sistemas operacionais móveis	7
Figura 3 – Tela inicial do iOS, versão 9	8
Figura 4 – Tela inicial do Windows Phone, versão 10	9
Figura 5 – Gráfico de uso das linguagens de programação	10
Figura 6 – Gráfico comparativo de popularidade entre os sistemas gerenciadores de banco de dados	13
Figura 7 – Telas de acesso e registro de usuário do aplicativo	17
Figura 8 – Tela principal e o menu de opções	18
Figura 9 – Telas referentes à funcionalidade ficha de exercícios	19
Figura 10 – Telas referentes à funcionalidade ficha de exercícios, a parte de gerenciar os exercícios	20
Figura 11 – Telas referntes à funcionalidade treinos	21
Figura 12 – Telas referentes à funcionalidade avaliação	22
Figura 13 – Diagrama de fluxo do funcionamento <i>offline</i>	24
Figura 14 – Trecho de código da interface FichaService utilizada pelo aplicativo . . .	25
Figura 15 – Trecho de código dos métodos de sincronização	26
Figura 16 – Trecho de código do métodos do servidor	27
Figura 17 – Modelo Entidade Relacionamento do aplicativo	28
Figura 18 – Diagrama de funcionalidades do aplicativo	29
Figura 19 – Representação ilustrativa da estrutura de funcionamento do sistema. . .	32

Lista de Quadros

Quadro 1 – Controle de versionamento e configuração do aplicativo desenvolvido. . 35

Lista de Abreviaturas e Siglas

AJAX	Asynchronous JavaScript and XML
API	Application Program Interface
CSS	Cascading Style Sheets
DER	Diagrama Entidade Relacionamento
GNU	GNU's Not Unix
HQL	Hibernate Query Language
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IIS	Internet Information Services
IP	Internet Protocol
JSON	JavaScript Object Notation
MER	Modelo Entidade Relacionamento
MVC	Model-View-Controller
ORM	Object-Relational Mapping
REST	Representational State Transfer
SQL	Structure Query Language
XML	Extensible Markup Language

Sumário

1 – Introdução	1
1.1 Apresentação	1
1.2 Objetivos	1
1.2.1 Objetivo principal	1
1.2.2 Objetivos específicos	2
1.3 Justificativa	2
1.4 Organização do Trabalho	3
2 – Revisão Bibliográfica	4
2.1 Conceitos	4
2.1.1 Tecnologias de informação	4
2.1.2 Dispositivos móveis	4
2.1.3 Aplicativos móveis	5
2.1.4 Android	6
2.1.5 iOS	7
2.1.6 Windows Phone	8
2.1.7 Java	9
2.1.8 PhoneGap	10
2.1.9 Xamarin	11
2.1.10 Web servers	11
2.1.11 Spring	12
2.1.12 Hibernate	12
2.1.13 MySQL	12
2.1.14 RESTful	14
2.1.15 HTTP	14
2.1.16 SQL	15
2.2 Trabalhos Relacionados	15
3 – Resultados	17
3.1 Parte externa	17
3.2 Parte interna	18
3.2.1 Fichas de Exercícios	19
3.2.2 Treinos	21
3.2.3 Avaliações Físicas	22
3.3 Funcionamento <i>Offline</i>	22
3.4 Disponibilidade de Dados	24

3.5	Comunicação entre Aplicativo e Servidor	25
3.6	Modelo Entidade Relacionamento	28
3.7	Diagrama de funcionalidades	29
4	– Metodologia	31
4.1	Ambiente de desenvolvimento	31
4.2	Estrutura do sistema	32
4.3	Documento de requisitos	34
4.4	Controle de Versionamento e Configuração	34
5	– Conclusões	36
	Referências	37
	Anexos	41
	ANEXO A – Documento de Requisitos	42
	ANEXO B – Estrutura Descritiva do Banco de Dados	55

1 Introdução

1.1 Apresentação

Nos últimos anos estamos vivenciando um período de intenso crescimento nas tecnologias de informação. Essas tecnologias vem sendo cada vez mais utilizadas no dia a dia das pessoas e estão ganhando mais espaço no mercado. Elas conseguem auxiliar em várias áreas, tais como saúde, educação, economia e negócios, política, entretenimento e lazer, dentre muitas outras, oferecendo soluções computacionais mais eficientes, seguras e automatizadas.

Uma das tecnologias de informação que estão se popularizando muito ultimamente são os dispositivos móveis, como por exemplo os *smartphones* e *tablets*. De acordo com uma pesquisa realizada pela IDC (IDC, 2016), a venda de *smartphones* cresce 13% ao ano, e atingiu 341,5 milhões de vendas por todo o mundo durante o segundo semestre de 2015. Devido ao aumento do uso dessa tecnologia nos últimos anos, também houve um crescimento na demanda por pequenas aplicações que oferecem soluções simples para vários tipos de problemas. Essas aplicações são conhecidas como aplicativos móveis.

As pessoas costumam levar seus *smartphones* para todos os lugares onde frequentam, fazem uso de diversos aplicativos móveis para fins diferentes. O que está se tornando cada vez mais comum é o uso desses dispositivos móveis nas academias durante os treinos (G1, 2013). As academias são ambientes que oferecem uma estrutura organizada para prática de atividades relacionadas à saúde física. Existem algumas operações que ainda são feitas de forma manual nesses ambientes, como por exemplo o gerenciamento de fichas de exercícios dos alunos, o preenchimento das fichas de avaliações físicas, cálculo do IMC, cálculo de gasto calórico, dentre alguns outros.

Por serem atividades simples que não apresentam lógicas de negócio complexas, a utilização de um aplicativo seria uma alternativa viável que poderia proporcionar uma maior facilidade e auxiliar na realização de tais atividades. Com isso, o foco deste trabalho é propor o desenvolvimento de um aplicativo móvel para academias.

1.2 Objetivos

1.2.1 Objetivo principal

Este trabalho tem como principal objetivo fazer um estudo sobre a metodologia de criação de aplicativos móveis e desenvolver um aplicativo voltado para o ambiente das

academias.

1.2.2 Objetivos específicos

De forma conjunta com o objetivo principal, alguns objetivos específicos podem ser citados, tais como:

- Discutir quais são as últimas tendências tecnológicas na área de desenvolvimento de aplicativos móveis.
- Abordar os processos de desenvolvimento de *software*, abrangendo um modelo de documentação.
- Desenvolver um protótipo de um aplicativo para auxiliar na montagem de ficha de exercícios e na execução dos treinos nas academias.
- Apresentar um caso de estudo de implementação de aplicações móveis.

1.3 Justificativa

Nos últimos anos, o uso de dispositivos móveis vem aumentando cada vez mais (AMADOR, 2015). Praticamente, quase todas as pessoas hoje em dia possuem um *smartphone* e fazem uso desse dispositivo em diversos lugares. Podemos perceber, ao andar pelas ruas da cidade, ao almoçar em um restaurante ou até mesmo em uma reunião empresarial, sempre há alguém utilizando um dispositivo móvel. Essa tecnologia acabou se tornando indispensável, e por isso, as empresas de desenvolvimento de *software* passaram a focar um pouco mais no desenvolvimento de aplicativos (IT Forum 365, 2015).

Com o aumento do uso de dispositivos móveis, é notório a popularização do uso de aplicativos. As pessoas fazem uso de vários aplicativos de diferentes finalidades, de acordo com o seu interesse e suas necessidades particulares. Existem aplicativos simples, como por exemplo, um aplicativo de gerenciamento de listas de tarefas, como até mesmo aplicativos mais complexos, tais como o aplicativo de um banco, onde são feitas consultas e transações que envolvem algo mais bem elaborado e necessita de uma maior segurança.

As academias estão sempre fazendo o gerenciamento das fichas de exercício dos seus alunos e o acompanhamento físico através de avaliações físicas. As fichas de exercício, na maioria das vezes, são gerenciadas manualmente fazendo o uso de papéis. O treinador ou professor é responsável por montar e fazer alterações nas fichas de seus alunos, o que gera um trabalho braçal considerável, já que o número de alunos das academias é grande, e as fichas precisam ser alteradas sempre, aproximadamente a cada dois meses.

Da mesma forma acontece com as avaliações físicas. Muitas vezes o avaliador físico faz todas as anotações manualmente, o que pode ser mais propício a erros humanos, já que envolve cálculos com medidas corporais.

Existem academias que fazem uso de sistemas computacionais para fazer o gerenciamento das fichas e o acompanhamento físico dos alunos. Porém, ainda existem os gastos com papéis para fazer a impressão dessas fichas e resultados de avaliações para entregar aos alunos.

Em resumo, a razão principal que motiva a realização deste trabalho é o interesse em usufruir de uma tendência tecnológica de sistemas de informação que possa gerir uma solução alternativa simples. Essa solução visa atender a demanda do gerenciamento de ficha de exercícios e avaliações físicas nas academias. Essa solução também visa aumentar a eficiência como esses processos são realizados, diminuir o trabalho braçal, eliminar o gasto com papéis e impressões, oferecer uma confiabilidade maior nos cálculos de medidas corporais e oferecer uma ferramenta intuitiva que tanto professores, avaliadores físicos e alunos sintam prazer em usar no dia a dia durante os treinos.

1.4 Organização do Trabalho

No capítulo de Introdução (1) é feita uma simples apresentação (1.1) que contém a contextualização do tema a ser abordado, mostra quais são os objetivos (1.2) que pretendem ser alcançados, juntamente com uma justificativa (1.3) para a realização de tal projeto.

Logo em seguida, o capítulo de Revisão Bibliográfica (2) traz alguns conceitos (2.1) que são fundamentais para a compreensão a cerca do desenvolvimento do aplicativo. Também mostra o estudo de alguns trabalhos relacionados (2.2) e quais foram os resultados alcançados por eles.

Depois, há o capítulo de Resultados (3), onde é feita uma apresentação do produto que foi desenvolvido, exibindo alguns exemplos de telas e funcionalidades.

Posteriormente, no capítulo de Metodologia (4), é apresentada toda a estrutura do sistema, mostra quais foram os documentos produzidos para o projeto, como foi feita a construção utilizando as tecnologias de desenvolvimento de aplicativos móveis.

Por fim, há o capítulo de Conclusão (5) onde são feitas as considerações finais a cerca do trabalho, ressaltando quais foram as dificuldades e propõe futuras melhorias para o produto desenvolvido.

2 Revisão Bibliográfica

Neste capítulo é abordada toda a revisão bibliográfica do trabalho, ou seja, todo o fundamento que foi tomado como base para a realização desse projeto. Ele está dividido em duas seções principais: Conceitos (seção 2.1) e Trabalhos Relacionados (seção 2.2).

2.1 Conceitos

Esta seção exibe pequenos conceitos que são fundamentais para a compreensão de alguns termos e tecnologias utilizadas neste trabalho.

2.1.1 Tecnologias de informação

A Tecnologia de Informação pode ser definida como um recurso computacional que tem como principal objetivo proporcionar atividades e funcionalidades capazes de obter, processar, armazenar, acessar e gerenciar informações (ALECRIM, 2015).

As informações são um tipo de patrimônio e agregam um certo valor para uma pessoa, uma empresa, ou qualquer outra entidade que precisa daquela informação para alcançar um objetivo. As empresas geram grandes massas de dados e informações diariamente e esse fluxo de dados precisa ser gerenciado por uma tecnologia computacional para que haja uma melhor confiabilidade, uma maior segurança e mais agilidade de processamento e análise.

As tecnologias de informações facilitam não apenas as empresas e instituições, mas também as pessoas que a utilizam no dia a dia, gerando informações constantemente em redes sociais, sites de compras, buscas na internet, dentre outras fontes de troca de informações.

Os dispositivos móveis são um tipo de tecnologia de informação que vem sendo tendência na vida das pessoas. Essa tecnologia é explicada com mais detalhes na subseção 2.1.2.

2.1.2 Dispositivos móveis

Antigamente, os computadores eram máquinas enormes que ocupavam grande quantidade de espaço e eram utilizados apenas por instituições de ensino, empresas e órgãos do governo (ALECRIM, 2015). Com a evolução das tecnologias, os computadores ficaram cada vez menores, mais eficientes, práticos e fáceis de usar, permitindo, inclusive,

serem levados para vários lugares e utilizados por qualquer pessoa. As tecnologias capazes de proporcionar essa maior flexibilidade são conhecidas como dispositivos móveis.

Um dispositivo móvel é um tipo de dispositivo computacional que tem como principais características a portabilidade, a compactabilidade e fácil manuseio (LEE; SCHNEIDER; SCHELL, 2005). O avanço das tecnologias de processamento, armazenamento e visualização proporcionou a criação desse tipo de dispositivo que se assemelha aos computadores tradicionais em questão de funcionalidade e uso.

Os dispositivos móveis são pequenos suficientes para caber na palma de uma mão. Temos como exemplos de dispositivos móveis, principalmente, os *smartphones* e os *tablets*. Esses dispositivos facilitam a vida das pessoas, pois eles podem ser utilizados em qualquer lugar, sem haver a necessidade de estarem presos por cabos.

De acordo com Ricardo R. Lecha (LECHETA, 2013), o mercado de dispositivos móveis vem crescendo cada vez mais nos últimos anos. Em 2013 havia mais de 3 bilhões de pessoas que utilizam aparelhos móveis, o que representava mais ou menos a metade da população mundial, de acordo com estudos.

Uma notícia mais recente do site Convergência Digital (GROSSMANN, 2015) mostra que em 2015 o mundo possuía mais de 7,1 bilhões de dispositivos móveis com linhas ativas, segundo estudos realizados pela TeleGeography.

Com esse crescimento no uso de dispositivos móveis, é importante ressaltar que também há um aumento na demanda por soluções de *software* que operam sobre essa tecnologia. Essas soluções são chamadas de aplicativos móveis, e o seu conceito é abordado com mais detalhes na subseção 2.1.3.

2.1.3 Aplicativos móveis

Os aplicativos móveis são *softwares* que utilizam de um dispositivo móvel para realizar alguma função para o usuário que está utilizando o dispositivo (LECHETA, 2013). Eles são nada mais do que programas simples que oferecem determinadas funções e que podem ser baixados e acessados dentro de um dispositivo móvel.

Os aplicativos são desenvolvidos com o propósito de facilitar a vida das pessoas oferecendo um serviço simples que pode ser feito com apenas alguns cliques. Existem aplicativos para comunicação, gerenciamento de dados, realização de compras, acesso a redes sociais, dentre outros. Cada dispositivo móvel provê uma plataforma diferente para que aplicativos possam ser instalados e executados. As plataformas mais comuns do mercado hoje em dia são as plataformas Android, iOS e Windows Phone. Esses sistemas operacionais são abordados com mais detalhes nas subseções 2.1.4, 2.1.5 e 2.1.6, respectivamente.

2.1.4 Android

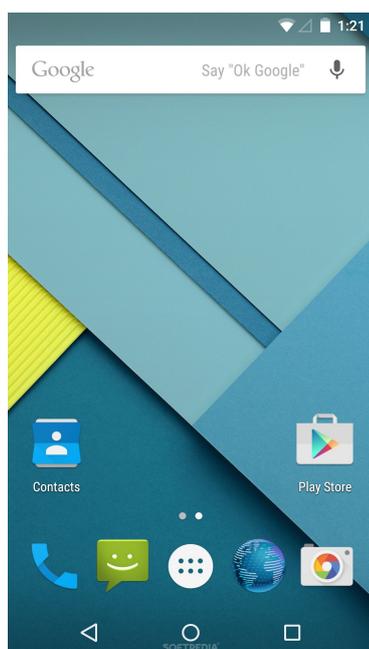
O Android (GOOGLE, 2016) é um sistema operacional inicialmente desenvolvido pela Google, e atualmente é mantido pela Open Handset Alliance, um grupo de empresas cujo objetivo é a criação de padrões abertos para sistemas de dispositivos móveis.

Esse sistema tem como características fundamentais o código fonte aberto. Ele é baseado no núcleo do Linux (linux.com, 2015), e os seus aplicativos são desenvolvidos utilizando a linguagem Java (abordada na subseção 2.1.7), o que permite uma maior facilidade de customização dos fabricantes de dispositivos móveis e desenvolvedores de aplicativos.

O Android funciona basicamente como qualquer outro sistema operacional para dispositivos móveis, oferecendo funções de gerenciar os processos dos aplicativos e manipular os recursos de hardware presentes no dispositivo. Um dos pontos interessantes é que o Android permite que o usuário associe sua conta do Google ao dispositivo e possa ter acesso aos seus dados nos aplicativos padrões do sistema. Ele conta com o Google Play Store, que é uma loja virtual onde o usuário pode baixar e instalar outros aplicativos desenvolvidos por terceiros.

Na Figura 1 temos o exemplo de duas telas de *smartphone* utilizando o sistema operacional Android Lollipop (5.0). A Figura 1(a) representa a tela inicial do sistema, já a Figura 1(b) representa a tela de menu de aplicativos.

Figura 1 – Exemplos de telas do sistema operacional Android, versão Lollipop (5.0)



(a) Tela inicial



(b) Tela do menu de aplicativos

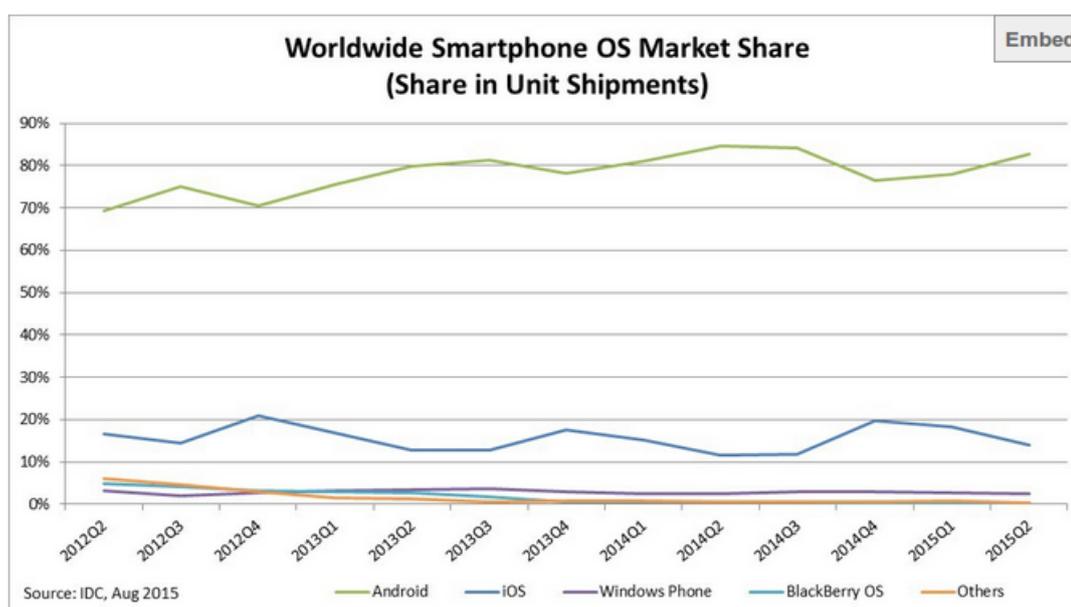
Fonte: (VASILE, 2014)

O Android opera principalmente em dispositivos móveis como *smartphones* e *tablets*. Esse sistema também é utilizado em outros dispositivos, tais como *notebooks*, TVs (Android TV), relógios de pulso (Android Wear), carros (Android Auto), *consoles* e câmeras digitais.

O Android é o sistema operacional mais utilizado atualmente. De acordo com a empresa de pesquisa e consultoria de tecnologia da informação Gartner (GARTNER, 2016), em 2014 o Android representava 76,0% do número de dispositivos adquiridos no mercado, enquanto o seu concorrente iOS detinha 20,4% e o Windows Phone apenas 2,8% do mercado. De acordo com a mesma referência, em 2015, esse número cresceu ainda mais, levando o Android a representar 80,7% do mercado de smartphones, enquanto o iOS e Windows Phone representavam 17,7% e 1,1%, respectivamente.

Outra pesquisa realizada em 2015 pela International Data Corporation (IDC, 2015) revela que o Android vem crescendo cada vez mais nos últimos anos, de 2012 até 2015 ele conseguiu passar de 69.3% para 82.8% no número de dispositivos utilizando o sistema operacional. A Figura 2 mostra um gráfico comparativo com os diferentes sistemas operacionais móveis do mercado.

Figura 2 – Gráfico comparativo sistemas operacionais móveis



Fonte: (IDC, 2015)

2.1.5 iOS

O iOS (APPLE, 2016) é um sistema operacional criado pela empresa Apple e seu sistema roda exclusivamente nos dispositivos móveis desenvolvido pela própria empresa. Dentre esses dispositivos podemos citar o iPhone, iPod Touch, iPad e Apple TV.

Assim como o Android e outros sistemas operacionais para dispositivos móveis, a

interface do sistema iOS é baseada na manipulação direta, ou seja, as interações com o sistema são feitas através de gestos e toques na tela do dispositivo. Existem sensores e acelerômetros que conseguem identificar as interações realizadas pelo usuário, para que o sistema interprete a ação e execute os eventos necessários, como por exemplo rotacionar o dispositivo, deslizar o dedo sobre a tela, fazer um movimento de “pinça” para ampliar ou diminuir uma imagem.

O usuário tem a opção de baixar novos aplicativos para o sistema através da App Store (App Store, 2016), uma loja virtual de aplicativos exclusivos da Apple, contém tanto aplicativos padrões quanto aplicativos desenvolvidos por terceiros. A Figura 3 mostra o exemplo de uma tela de *smartphone* utilizando o sistema operacional iOS 9.

Figura 3 – Tela inicial do iOS, versão 9



Fonte: (VILLAS-BOAS, 2015)

2.1.6 Windows Phone

O Windows Phone (MICROSOFT, 2016b) é um sistema operacional para dispositivos móveis criado pela empresa Microsoft pra substituir o antigo Windows Mobile de 2000. Esse sistema foi lançado em outubro de 2010 e tem como principal linha de aparelhos alvo o Lumia da Nokia, recentemente comprada pela Microsoft.

O objetivo da Microsoft é criar um sistema que seja capaz de servir como uma única plataforma, tanto para computadores, *notebooks*, *smartphones*, *tablets*, e até mesmo o Xbox. A idéia é fazer uma integração parecida com o que a Google faz com o Android e a

Apple faz com o iOS.

A Figura 4 mostra o exemplo de uma tela de *smartphone* utilizando o sistema operacional Windows Phone 10.

Figura 4 – Tela inicial do Windows Phone, versão 10



Fonte: (VASILE, 2015)

Por chegar no mercado recentemente, quando os seus concorrentes já estavam bem consolidados, o Windows Phone ainda não abrange um grande número de usuários. Outro fator é que a sua loja virtual de aplicativos Windows Phone Store não tem uma grande variedade de aplicativos como as demais, conta apenas com aplicativos mais populares como Twitter (TWITTER, 2016), Facebook (FACEBOOK, 2016), Instagram (INSTAGRAM, 2016).

2.1.7 Java

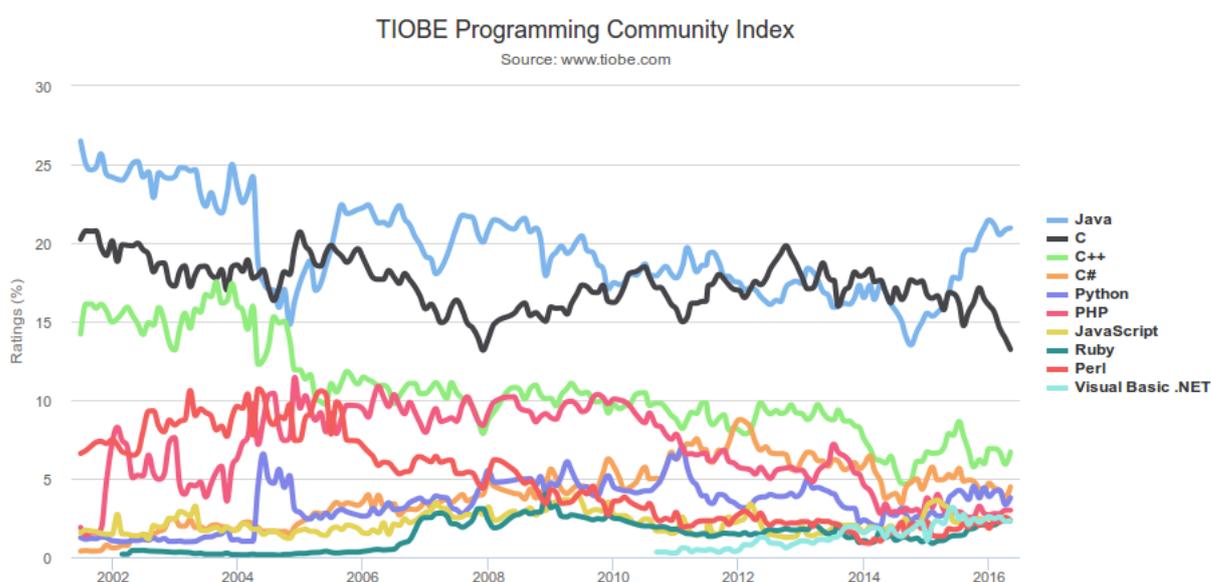
O Java (ORACLE, 2016) é uma linguagem de programação de alto nível que utiliza o paradigma de orientação a objetos. Ela foi desenvolvida originalmente pela empresa Sun Microsystems e atualmente pertence à empresa Oracle. Essa linguagem é simples, robusta, eficiente, segura e confiável. A linguagem apresenta código fonte aberto, liberado sobre a licença GNU General Public License (GNU, 2016).

Um dos principais benefícios que essa linguagem oferece é a portabilidade. Todo código fonte é compilado, e no final de tudo é gerado um arquivo *bytecode*, esse arquivo pode ser executado em uma máquina virtual. Isso significa que o desenvolvedor não precisa

escrever códigos diferentes para cada sistema operacional, ele pode simplesmente escrever uma única vez, e executar em qualquer lugar que tenha o Java Virtual Machine instalado.

Atualmente o Java é a linguagem de programação mais utilizada no mundo. Sua utilização é feita principalmente para desenvolvimento de aplicações *web*, aplicações *desktop* e criação de aplicativos móveis para Android. De acordo com a empresa de qualidade de *software* TIOBE (TIOBE, 2016), em maio de 2016 o Java foi a linguagem mais requisitada por ferramentas de buscas na internet, ocupando aproximadamente 20,956% do total das pesquisas realizadas.

Figura 5 – Gráfico de uso das linguagens de programação



Fonte: (TIOBE, 2016)

A Figura 5 mostra um gráfico comparativo das linguagens de programação mais utilizadas e mais referenciadas durante os últimos anos. Podemos perceber que o Java já está bem consolidado no mercado. No gráfico ele quase sempre apresenta maior relevância entre as demais linguagens por ser eficiente e proporcionar os recursos necessários para o desenvolvimento de qualquer tipo de *software* em diversas plataformas.

2.1.8 PhoneGap

PhoneGap é um *framework* para o desenvolvimento de aplicações móveis pertencente à Adobe Systems (Adobe Systems Inc., 2016). Esse *framework* permite que o desenvolvedor crie aplicações móveis apenas utilizando tecnologias do mundo *web*, tais como HTML, CSS e Javascript.

Através do PhoneGap o programador pode criar aplicações híbridas, ou seja, que é

compatível com várias plataformas sem se preocupar em aprender API's de plataformas específicas, como por exemplo Android, iOS e Windows Phone.

2.1.9 Xamarin

Xamarin (Microsoft Corporation, 2016b) é uma empresa pertencente à Microsoft que oferece uma plataforma de desenvolvimento de aplicações móveis. A empresa foi fundada em maio de 2011 por engenheiros responsáveis pela criação do Mono, Mono for Android e MonoTouch, que são implementações multi-plataforma da Microsoft .NET.

O projeto Xamarin fornece ferramentas para escrever aplicativos nativos para Android, iOS e Windows Phone através de um único código fonte. Os desenvolvedores utilizam a linguagem C# para criar suas aplicações multi-plataforma e podem exportar suas aplicações para qualquer plataforma através das funcionalidades presentes no Xamarin. O projeto Xamarin também disponibiliza opções para fazer testes automatizados. De acordo com o site oficial (Microsoft Corporation, 2016a), o Xamarin é utilizado por mais de 1 milhão de desenvolvedores no mundo inteiro.

2.1.10 Web servers

Os *Web servers* são computadores hospedados na rede que utilizam do protocolo HTTP (subseção 2.1.15) para oferecer um serviço de entrega de páginas *web* (ROUSE, 2015). Esses servidores têm um IP fixo, geralmente podem ser acessados por um nome de domínio, como por exemplo *www.exemplo.com.br*.

A principal função dos *Web servers* é oferecer um serviço baseado na arquitetura cliente-servidor, onde os computadores clientes fazem solicitações ao computador servidor, este servidor gerencia essas requisições recebidas, processa de acordo com suas métricas e retorna a resposta para o computador que realizou aquela requisição.

Qualquer computador pode se tornar um servidor apenas conectando a máquina à internet e instalando um *software* que oferece um serviço de servidor. Dentre esses *softwares* de servidor os mais conhecidos são o Apache (apache.org, 2016), Nginx (NGINX Inc., 2016) e Microsoft's Internet Information Server (IIS) (MICROSOFT, 2016a). De acordo com uma pesquisa realizada pela equipe W3Techs.com (W3TECHS, 2016), o Apache ocupa 52,4% do número de servidores em funcionamento na internet.

Os *Web servers*, em geral, entregam páginas *web* no formato HTML para serem exibidas nos *browsers* dos usuários. Entretanto, também pode ser utilizados para oferecer respostas em diferentes formatos, como por exemplo XML (WALSH, 2016) e JSON (JSON.ORG, 2016). Esses outros formatos provêm uma padronização simples que facilita na comunicação entre o servidor e aplicações mais leves, como por exemplo requisições AJAX e até mesmo requisições solicitadas por aplicativos móveis. O AJAX é um conjunto

de técnicas utilizadas na programação *web* para carregar informações na página de forma assíncrona, ou seja, sem a necessidade de carregar a página novamente (Mozilla Developer Network, 2016).

Uma das coisas mais comuns no desenvolvimento de aplicativos é a integração com um *Web server* para fazer consultas e armazenamento de dados, deixando todas as tarefas de processamento pesado por conta do servidor, assim tornando os aplicativos mais leves.

2.1.11 Spring

Spring (Pivotal Software, 2016) é um *framework* de código fonte aberto para a desenvolvimento de aplicações Java. Esse *framework* se tornou amplamente popular entre a comunidade Java por ter um alto desempenho, fácil testabilidade e reusabilidade de código. Seus principais recursos podem ser utilizados em qualquer tipo de aplicação Java, inclusive para criação de aplicações *web*.

Esse *framework* apresenta como principais características injeção de dependência, programação orientada a aspectos, suporte para aplicações *web* na arquitetura MVC, *web services* RESTful, dentre outras.

2.1.12 Hibernate

Hibernate (hibernate.org, 2016) é um *framework* de código fonte aberto para o mapeamento objeto-relacional em aplicações Java. Ele oferece funções para fazer o mapeamento de um modelo de orientação a objetos para um banco de dados relacional. Esse *framework* facilita a execução de *queries* e recuperação de dados, reduzindo significativamente o tempo de desenvolvimento das aplicações. Além disso, ele permite escrever *queries* na sua própria extensão SQL portátil chamada Hibernate Query Language (HQL).

No Hibernate, as classes podem ser mapeadas para tabelas no banco de dados através de um arquivo de configuração XML ou até mesmo utilizando Java Annotations, uma espécie de meta dados sintático que é adicionado ao código fonte Java para que possam ser processadas em tempo de execução. Dessa forma, o Hibernate consegue criar a estrutura no banco e realizar a persistência dos dados de forma fácil, poupando o trabalho que o programador teria para fazer isso da forma tradicional.

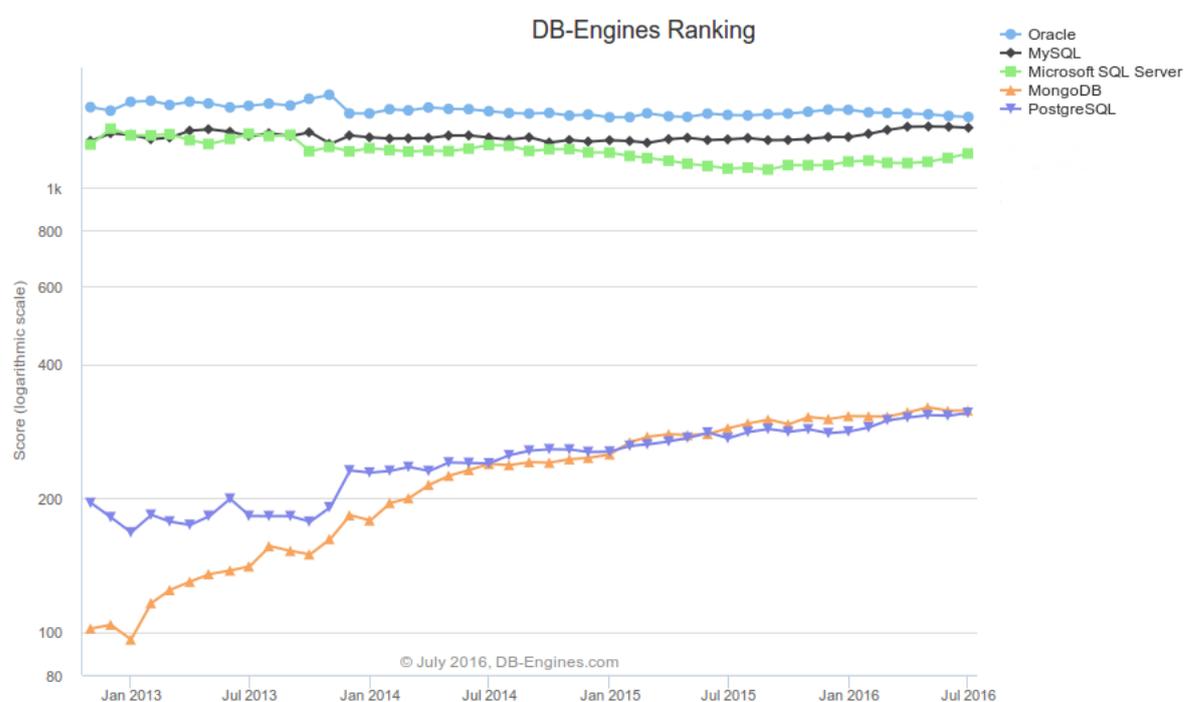
2.1.13 MySQL

O MySQL (Oracle Corporation, 2016) é um sistema gerenciador de banco de dados relacional e de código fonte aberto. Esse sistema utiliza a linguagem SQL para selecionar, inserir, alterar, excluir e gerenciar todos os registros presentes em uma base de dados.

Ele foi desenvolvido originalmente pela empresa sueca MySQL AB em 1995. Atualmente o MySQL pertence à empresa Oracle. Por ser um sistema gerenciador de banco de dados gratuito, ele oferece uma possibilidade econômica para aplicações que utilizam banco de dados. Por isso, atualmente, ele é um dos principais componentes utilizados para criação de aplicações públicas na internet, sendo utilizado até mesmo por grandes empresas, tais como NASA, Sony, Bradesco, dentre outras.

De acordo com um *ranking* publicado em um site de pesquisa sobre a popularidade dos sistemas gerenciadores de banco de dados (DB-ENGINES, 2016b), em julho de 2016 o MySQL foi considerado o segundo sistema gerenciador de banco de dados mais popular, seguido por Microsoft SQL Server, MongoDB e PostgreSQL.

Figura 6 – Gráfico comparativo de popularidade entre os sistemas gerenciadores de banco de dados



Fonte: (DB-ENGINES, 2016a)

O gráfico da Figura 6 mostra uma comparação do nível de popularidade entre os sistemas gerenciadores de banco de dados através de um *score*. Esse *score* é uma métrica criada pelo site baseada em diversos aspectos, tais como a frequência das buscas em sites de pesquisas, número de menções em fóruns de tecnologia, número de vagas de emprego, dentre outras. Desde janeiro de 2013, o MySQL vem apresentando um *score* entre os três maiores. Com isso, é possível perceber que esse sistema gerenciador de banco de dados é uma das tendências a serem escolhidas na hora do desenvolvimento de aplicações em geral.

Dentre outras características relevantes do MySQL, podemos citar a portabilidade, a compatibilidade com diversas linguagens de programação, facilidade de uso e aprendizado, não demanda grande quantidade de recursos de *hardware*, apresenta um alto desempenho e estabilidade.

2.1.14 RESTful

O REST ou RESTful, cuja sigla significa *Representational State Transfer*, em português Transferência de Estado Representacional, consiste em uma arquitetura para aplicações de *hipermídia* na rede. Essa arquitetura permite construir serviços *web* leves, de fácil manutenção e escaláveis.

De acordo com um livro “Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON” (KALALI; MEHTA, 2013), essa arquitetura pode ser definida pelas características abaixo.

- Associação dos ID's aos recursos, ou seja, cada recurso é identificado através de uma URI única.
- Uso de métodos HTTP padrões. Esses métodos definem quais serão os tipos de operações que serão realizadas. São exemplos de métodos HTTP os métodos POST, GET, PUT e DELETE.
- Múltiplos formatos de dados oferecidos por um recurso. Isso significa que um recurso pode ser obtido em diversos formatos diferentes, como por exemplo JSON e XML.
- Sem estado. As requisições HTTP trocadas entre cliente e servidor contêm todas as informações necessárias para a comunicação. Por isso, não há a necessidade de criar sessões e manter o estado das mesmas.

A arquitetura RESTful trabalha com recursos, e oferece todas as ferramentas necessárias para fazer a manipulação desses recursos. As aplicações atuais estão tendendo a utilizar a arquitetura RESTful como base de sua comunicação e estão aparecendo cada vez mais API's que são desenvolvidas à base dessa arquitetura.

2.1.15 HTTP

O HTTP (HyperText Transfer Protocol) é um protocolo de redes utilizado na *web* para definir como as mensagens são transmitidas e formatadas. Também cria uma padronização de como as ações devem ser interpretadas pelos *web servers* e pelos *browsers* para responder aos comandos solicitados. Esse protocolo é responsável por elaborar um conjunto de regras de como os arquivos devem ser transferidos na internet, sejam eles arquivos de texto, imagens, audios, vídeos, dentre outros arquivos de *multimídia* (ROUSE, 2006).

2.1.16 SQL

SQL é uma linguagem utilizada para acessar e manipular banco de dados relacionais através da realização de consultas. No SQL podemos adicionar, editar e deletar registros, modificar as estruturas e índices das tabelas, e realizar transações para processar um conjunto de operações (ROUSE, 2016).

2.2 Trabalhos Relacionados

Esta seção mostra alguns estudos e trabalhos relacionados que já foram realizados anteriormente e que possam contribuir para este trabalho de alguma forma.

Eric J. Topol (TOPOL, 2013), escritor e doutor em medicina formado na *University of Rochester School of Medicine*, critica em seu livro *The Creative Destruction of Medicine: How the Digital Revolution Will Create Better Health Care* que a medicina hoje em dia está com uma certa deficiência na precisão dos tratamentos e propõe que tecnologias digitais, principalmente as de dispositivos móveis, devem ser utilizadas para melhorar a qualidade e eficiência nos serviços de saúde oferecidos.

Em seu trabalho de conclusão de curso, Paris .L, Santos .T e Junior .V (PARIS; SANTOS; JUNIOR, 2015) propõem a criação de um aplicativo para ser utilizado dentro das academias, com a intenção de gerenciar fichas de exercícios e avaliações físicas, além de facilitar o acompanhamento dos treinos e diminuir o gasto com papéis. No final do trabalho, como resultado, eles realizaram a criação de um protótipo que consiste em um aplicativo utilizando a tecnologia PhoneGap, e esse aplicativo é integrado com um *website* utilizando a tecnologia PHP.

O artigo *Health and Fitness App Use in College Students: A Qualitative Study* (GOWIN et al., 2014) faz um estudo da utilização de aplicativos relacionados à saúde entre estudantes de faculdade que vivenciam obesidade e outros sérios problemas. Nesse artigo, a proposta foi tentar descobrir se o uso de aplicativos pode ajudar no desenvolvimento de rotinas de exercícios e influenciar uma melhora nos hábitos alimentares. Como resultado, a maioria dos estudantes que participaram do experimento alegaram que os aplicativos contribuíram de uma forma positiva para a adoção de novos comportamentos e melhora na saúde.

No artigo *Importância da Tecnologia no Acompanhamento das Atividades dos Alunos de Academia* (BENTO et al., 2013), é mencionado o fato de que o número de academias está crescendo em grande escala em todo o país. Porém, nem todas elas oferecem ferramentas para controle da evolução e acompanhamento de seus alunos. O aplicativo propõe uma pesquisa para identificar os tipos de pessoas que praticam academia, e qual o seu nível de interesse em utilizar Tecnologias de Informação para auxiliar no seu treino e fazer o

acompanhamento físico. Os resultados apontam que dos 131 alunos entrevistados, 95% deles demonstram interesse na implantação de tecnologias na sua rotina da academia.

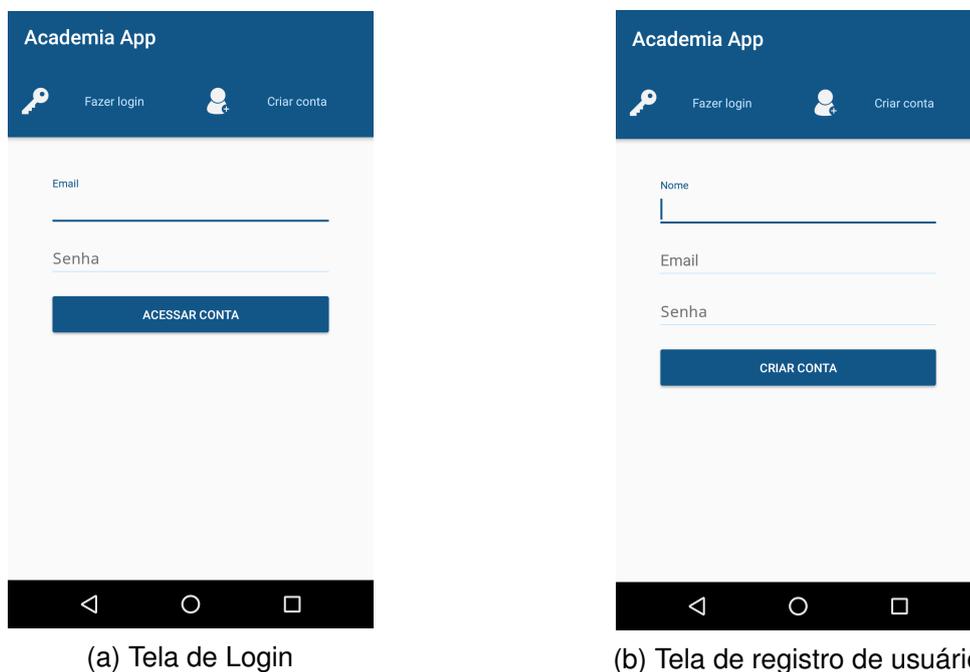
O artigo *Mobile Exercise Apps and Increased Leisure Time Exercise Activity: A Moderated Mediation Analysis of the Role of Self-Efficacy and Barriers* (LITMAN et al., 2015) propõe um estudo sobre como os aplicativos móveis de exercícios físicos podem ajudar a melhorar os níveis de saúde e os resultados dos exercícios. O objetivo do artigo foi fazer uma pesquisa entre alguns grupos de pessoas que utilizam ou não aplicativos relacionados à saúde. Como resultado, foi possível chegar a conclusão de que aproximadamente 75% das pessoas que utilizam aplicativos mostraram-se mais ativos e com melhores índices do que os que não utilizam. Além disso, os dados sugerem que os aplicativos funcionam como uma forma de estímulo para que os usuários possam ultrapassar barreiras através do desafio de superar suas próprias metas.

3 Resultados

Neste capítulo são apresentados os resultados obtidos após o desenvolvimento do aplicativo.

3.1 Parte externa

Figura 7 – Telas de acesso e registro de usuário do aplicativo



Fonte: Elaborado pelo autor

Na Figura 7 são ilustradas as telas da parte externa do aplicativo.

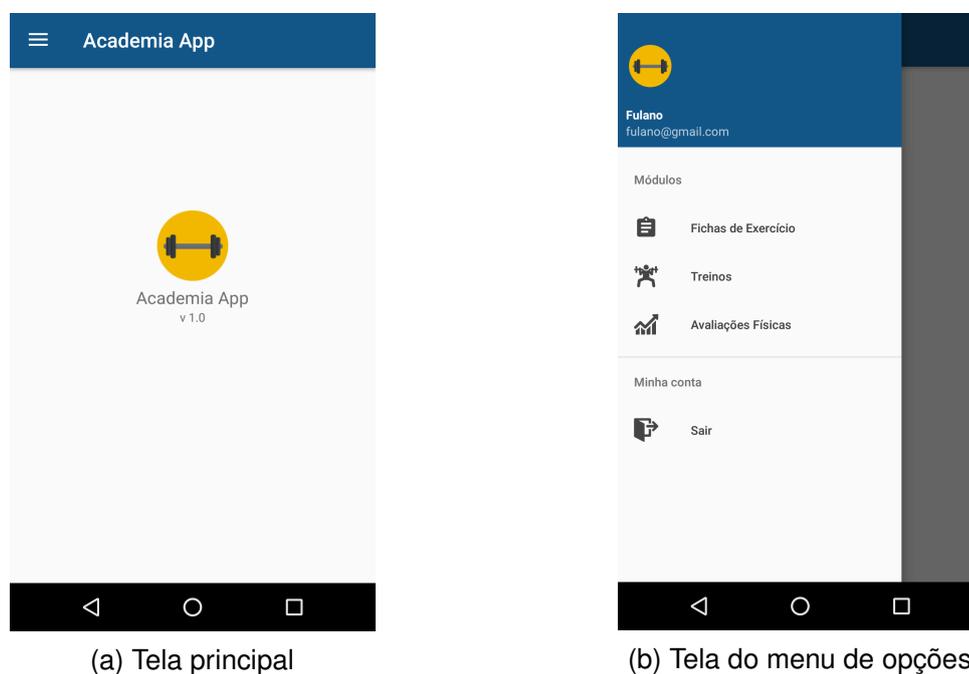
A tela de login, representada na Figura 7(a), é composta pelos campos onde o usuário deverá informar seus dados para ter acesso ao sistema.

Na Figura 7(b) temos a tela de registro de usuário no sistema e ela é composta pelos campos necessários para efetuar o cadastro de um usuário.

Ambas as telas, a partir das devidas validações, dão acesso à parte interna do sistema, onde estão presentes todas as outras funcionalidades.

3.2 Parte interna

Figura 8 – Tela principal e o menu de opções



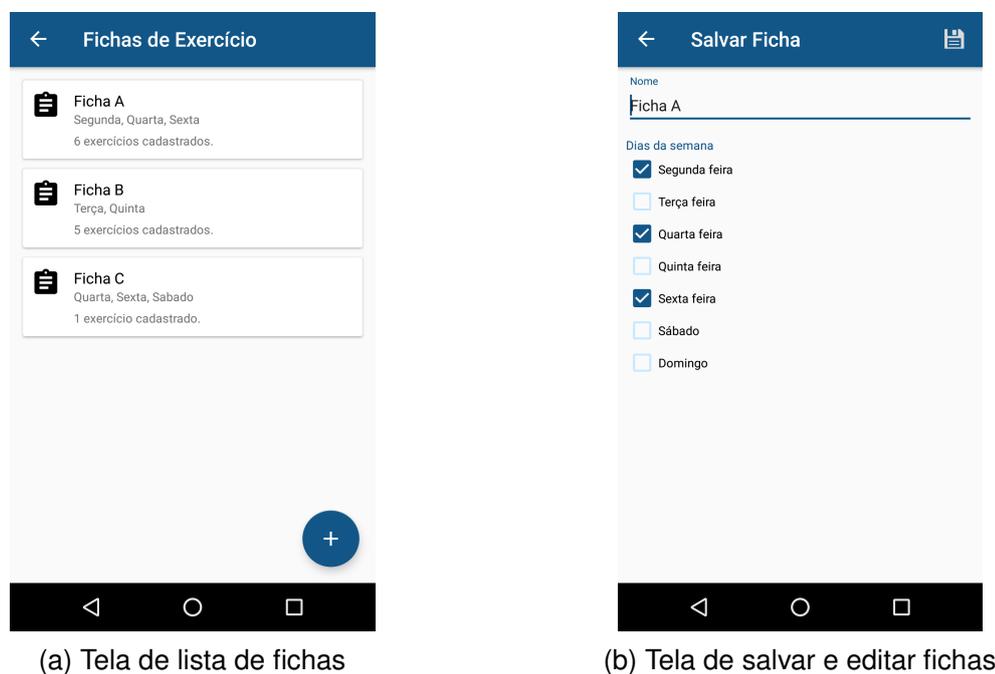
Fonte: Elaborado pelo autor

Na Figura 8 é exibida a tela principal da parte interna do sistema. Ao acessar a parte interna, a primeira tela que o usuário vê é a representada na Figura 8(a). Ao clicar no símbolo superior esquerdo ou arrastar o dedo na tela da esquerda para a direita, o menu de opções é aberto, como mostra a Figura 8(b).

No menu de opções, o usuário pode escolher qual funcionalidade ele deseja utilizar no momento. Ele também pode escolher a opção de sair da sessão atual para que ele possa voltar à parte externa e acessar uma outra conta.

3.2.1 Fichas de Exercícios

Figura 9 – Telas referentes à funcionalidade ficha de exercícios



Fonte: Elaborado pelo autor

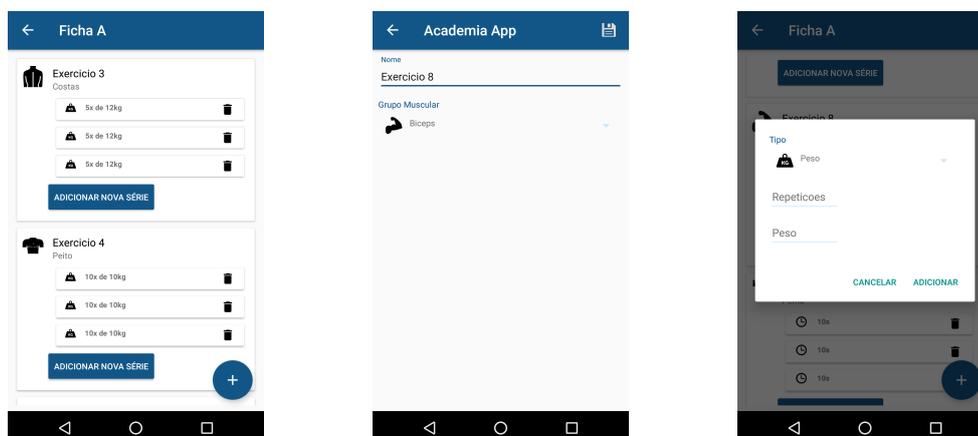
A Figura 9 mostra as telas do módulo de ficha de exercícios.

A tela de listagem de fichas, como mostra na Figura 9(a), é responsável por listar todas as fichas cadastradas no aplicativo. Através dessa tela o usuário pode visualizar e manipular as suas fichas.

No canto inferior direito dessa mesma tela há um botão azul com símbolo de inserção. Ao clicar nesse botão o usuário é direcionado para a tela de criar nova ficha de exercício, como mostra a Figura 9(b). Nessa tela há um formulário onde o usuário preenche as informações da nova ficha e efetua o cadastro.

Essa mesma tela é usada para editar as fichas já existentes. Na tela de listagem há a opção de excluir, basta apenas o usuário pressionar um registro, este é selecionado e o usuário pode clicar no símbolo de exclusão para efetuar a operação.

Figura 10 – Telas referentes à funcionalidade ficha de exercícios, a parte de gerenciar os exercícios



(a) Tela de lista de exercícios da ficha (b) Tela de salvar e editar exercícios (c) Tela de atrelar séries à um exercício

Fonte: Elaborado pelo autor

Ao clicar em uma ficha da listagem, o usuário é direcionado à tela de listagem de exercícios dessa ficha, como mostra a Figura 10(a). Essa tela lista todos os exercícios presentes na ficha, inclusive as séries que estão atreladas à esse exercício. Essa tela funciona de forma semelhante à tela de fichas, e através dela o usuário pode manipular os registros.

O usuário pode inserir um novo exercício na ficha através do botão de inserção, que irá abrir um formulário com os dados que precisam ser informados, como é representado na Figura 10(b).

Ao clicar no botão Adicionar Nova Série, uma janela abre para que o usuário forneça as informações da nova série, como é mostrado na Figura 10(c).

3.2.2 Treinos

Figura 11 – Telas referentes à funcionalidade treinos



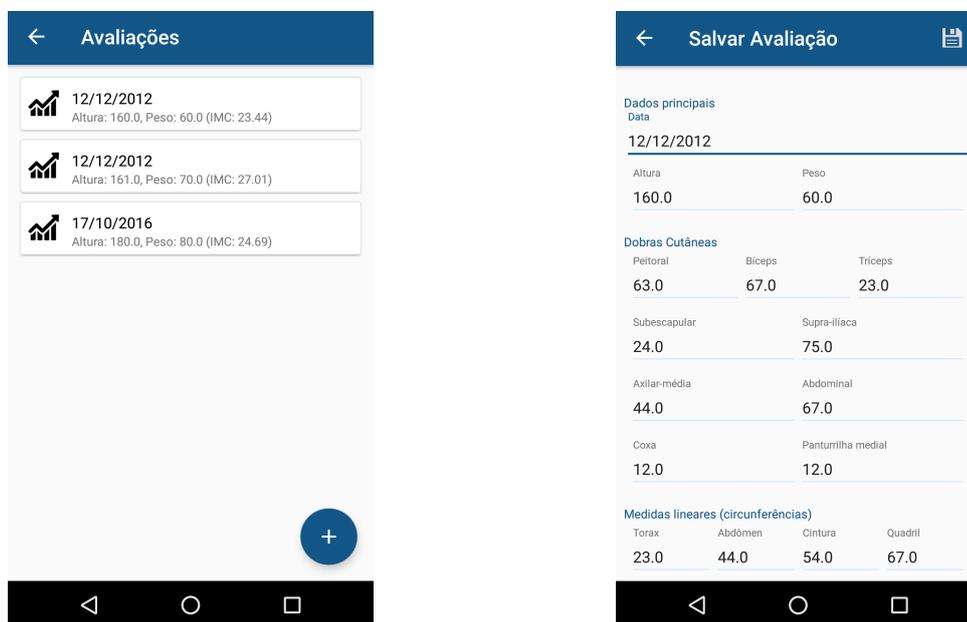
Fonte: Elaborado pelo autor

A Figura 11 mostra as telas do módulo de treino. As operações de listar, cadastrar, editar e excluir se comportam de forma semelhante às do módulo de ficha de exercícios.

Ao clicar em um treino, a tela de gerenciar exercícios será aberta, como mostra a Figura 11(c). Nessa tela, o usuário pode marcar quais exercícios já foram realizados no treino.

3.2.3 Avaliações Físicas

Figura 12 – Telas referentes à funcionalidade avaliação



(a) Tela de lista de avaliações

(b) Tela de salvar e editar avaliações

Fonte: Elaborado pelo autor

A Figura 12 mostra as telas do módulo de avaliações. Esse módulo se comporta da mesma forma que os módulos anteriores, oferecendo funções de listar, cadastrar, editar e excluir.

A tela de cadastro de avaliações, representada na Figura 12(b), é composta por um formulário que contém todos os campos necessários para registrar uma avaliação.

3.3 Funcionamento *Offline*

Hoje em dia, os dispositivos móveis conseguem ter um acesso fácil às redes *wifi* com conexão à internet. A maioria dos estabelecimentos que frequentamos tem um roteador com antena *wifi* que geram o sinal de internet para dispositivos móveis. Porém, nem todo lugar disponibiliza essa tecnologia, e muitas vezes podem ocorrer falhas na rede, o que impede que o dispositivo consiga se conectar e trocar informações na *web*.

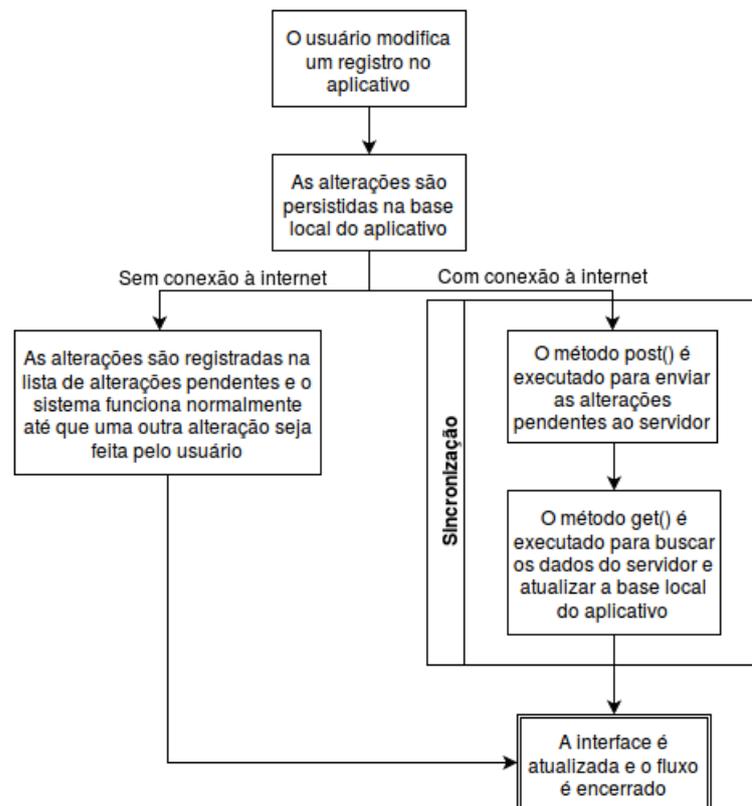
Levando em consideração que o recurso de acesso à internet nem sempre está disponível para os usuários do aplicativo, podemos dizer que há a necessidade de criar uma base de dados local, onde os dados do usuário ficam armazenados temporariamente até que o dispositivo consiga ter uma conexão com a internet para sincronizar os dados com o servidor.

Neste contexto, o servidor é responsável por receber essas atualizações feitas pelo usuário e garantir que os dados sempre estarão em conformidade. Esse é um dos diferenciais do aplicativo, o dispositivo não precisa estar conectado à internet para funcionar corretamente, sendo assim o usuário não fica impedido de realizar suas atividades caso esteja *offline*. Com essa técnica, conseguimos garantir que o aplicativo funcionará normalmente em quaisquer condições de rede.

Em seu trabalho de conclusão do mestrado, Guedes (GUEDES, 2015) propõe a criação de um *framework* para sincronização de dados *offline* para aplicações em Android, e além disso ele ressalta a importância de oferecer esse recurso de sincronização para tornar as aplicações mais confiáveis, acessíveis e sem interrupção.

O funcionamento *offline* é um dos grandes diferenciais desse aplicativo, já que grande parte dos aplicativos hoje em dia ainda são completamente dependentes de conexão com a internet para buscar seus dados e realizar suas operações, o que muitas vezes impede o usuário de usufruir das funcionalidades presentes dentro do sistema quando não há conexão com a internet.

Para compreender melhor como funciona esse funcionamento *offline*, foi criado o diagrama da Figura 13. De acordo com esse diagrama, inicialmente todas as alterações feitas pelo usuário são persistidas na base de dados local do aplicativo, e quando há conexão com a internet é realizada uma troca de dados com o servidor. Quando não há conexão com a internet, as alterações são registradas na lista de atualizações pendentes, e assim que uma nova alteração é feita com conexão à internet, todas as alterações pendentes são enviadas ao servidor.

Figura 13 – Diagrama de fluxo do funcionamento *offline*

Fonte: Elaborado pelo autor

Os detalhes de implementação são explicados de forma mais detalhada na seção 3.5, onde é feita uma demonstração do código fonte de algumas partes do sistema.

3.4 Disponibilidade de Dados

Outra característica que podemos perceber na estrutura representada na Figura 19 é que os dados ficam todos armazenados em um banco de dados principal.

Podem ocorrer situações em que o usuário terá a necessidade de migrar seus dados para outro dispositivo. Por exemplo, o usuário teve seu *smartphone* roubado, ou sentiu a necessidade de comprar um outro *smartphone*. Também pode acontecer de o usuário simplesmente desejar acessar sua conta no dispositivo de um amigo.

Para atender à essa necessidade de disponibilidade de dados, existe a presença do servidor principal. Ao acessar sua conta em um dispositivo diferente, todos os dados que estão presentes no servidor são copiados para o banco de dados local desse novo dispositivo. Com isso, o usuário é poupado do trabalho de ter que migrar manualmente a sua base de dados, o que poderia demandar um conhecimento técnico mais avançado. Também poupa o usuário de ter que recadastrar todos os registros, o que poderia gerar

uma certa frustração e um grande esforço dependendo da quantidade de dados.

Essa característica é um outro diferencial do aplicativo, já que grande parte dos aplicativos presentes hoje na Play Store armazenam seus dados apenas localmente, caso o usuário queira acessar esses dados de outro dispositivo, não será possível nesses outros aplicativos.

3.5 Comunicação entre Aplicativo e Servidor

A biblioteca Retrofit utilizada no Android e o *framework* Spring utilizado no servidor formaram uma ótima combinação para a troca de requisições RESTful, conseqüentemente proporcionando uma comunicação mais confiável entre o aplicativo e o servidor.

Primeiramente, no lado do aplicativo, foi necessário criar os métodos que irão solicitar algum serviço no servidor. A Figura 14 mostra o exemplo de uma interface, utilizando os padrões do Retrofit, contendo métodos que irão enviar as atualizações ao servidor e buscar os dados do mesmo.

Figura 14 – Trecho de código da interface FichaService utilizada pelo aplicativo

```
public interface FichaService {  
    @GET("ficha/listar/{id}") 1  
    Call<Status<List<Ficha>>> listar(@Path("id") String id); 2  
  
    @POST("ficha/update")  
    Call<Status> update(@Body List<UpdateFicha> updateFicha);  
}
```

Fonte: Elaborado pelo autor

Na Figura 14(1), temos uma anotação que especifica qual será a *url* que a biblioteca deverá solicitar em sua requisição *http*. Nesse caso, por exemplo, temos a *url* "ficha/listar/{id}", que irá retornar uma lista de fichas de acordo com o identificador do usuário. Logo em seguida, na Figura 14(2), temos a especificação do método que deverá ser implementado para essa regra da anotação, e esse método deverá tratar as respostas que serão retornadas do servidor.

A Figura 15 mostra como foram implementados os métodos *post()* e *get()*, mencionados no diagrama da Figura 13. Esses métodos são representados na Figura 15(1) e Figura 15(4), respectivamente.

Figura 15 – Trecho de código dos métodos de sincronização

```
@Override
protected void post() { 1
    SyncEvent.send(getSyncType(), SyncStatus.IN_PROGRESS);
    Log.i("FichasSync", "execute post method");
    List<UpdateFicha> updates = UpdateFichaDao.getAll(Realm.getDefaultInstance()); 2
    FichaService apiService =
        RestfulAPI.getClient().create(FichaService.class);
    Call<Status> call = apiService.update(updates); 3
    call.enqueue(new Callback<Status>() {...});
}

@Override
protected void get() { 4
    Log.i("FichasSync", "execute get method");
    FichaService apiService =
        RestfulAPI.getClient().create(FichaService.class);
    Call<Status<List<Ficha>>> call = apiService.listar(SessionUtils.getInstance(context).getIdUsuario());
    call.enqueue(new Callback<Status<List<Ficha>>>() {...}); 5
}
```

Fonte: Elaborado pelo autor

Nesse exemplo, a Figura 15(2) mostra a linha de código em que as atualizações feitas pelo usuário são coletadas, e logo em seguida na Figura 15(3) essas atualizações são enviadas ao método *update*. Nessa parte é feita a implementação dos métodos criados na interface *FichaService*. No final da implementação do método da Figura 15(3), é chamado o método *get()*. No ponto 5 também é feita a implementação de um método da interface citada anteriormente.

A Figura 16 mostra como foi desenvolvido o lado do servidor. Na Figura 16(1) temos a classe *FichaController*, com uma anotação que representa como a *url* será construída dentro daquela classe. Por exemplo, na Figura 16(2) temos a especificação do método *listar*, com uma anotação que diz que todas as requisições destinadas à *url* "ficha/listar/{id}" serão processadas por aquele método.

Figura 16 – Trecho de código do métodos do servidor

```
@Controller
@RequestMapping("/ficha")
public class FichaController { 1

    @Autowired
    FichaServices fichaServices;

    @RequestMapping(value = "/listar/{id}", method = RequestMethod.GET) 2
    public @ResponseBody Status listar(@PathVariable("id") String id) {
        try {
            List<Ficha> dados = fichaServices.listar(id);
            return new Status(1, "Fichas listadas com sucesso!", dados);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return new Status(0, "Houve um erro ao tentar realizar esta ação!", null);
    }

    @RequestMapping(value = "/update", method = RequestMethod.POST, consumes = MediaType.APPLICATION_JSON_VALUE)
    public @ResponseBody Status update(@RequestBody List<UpdateFicha> updates) { 3

        // ...

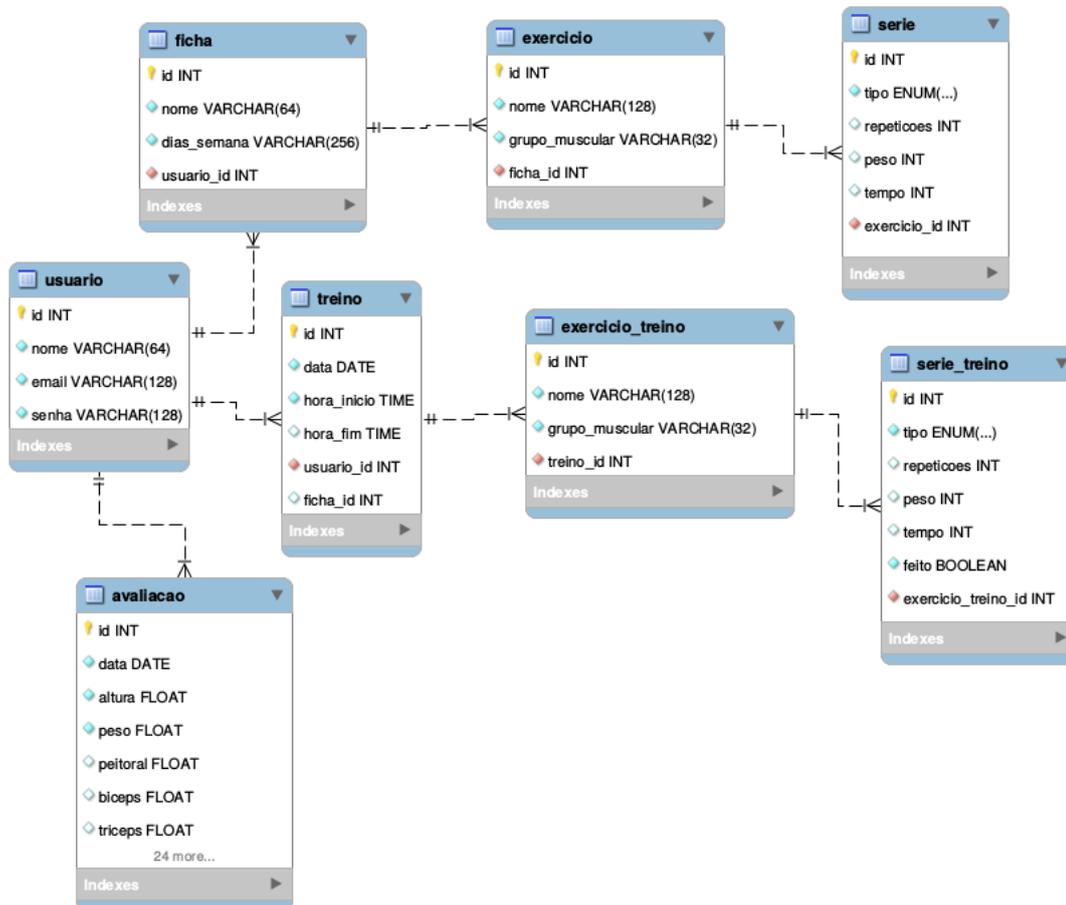
        return new Status(1, "Nenhum update a ser realizado!", null);
    }
}
```

Fonte: Elaborado pelo autor

Através dessa construção, conseguimos obter uma sintonia perfeita entre a biblioteca Retrofit e o *framework* Spring. Como o aplicativo e o servidor foram desenvolvidos utilizando a mesma linguagem de programação, foi fácil criar esse canal de comunicação utilizando a mesma interface de serviço e as mesmas anotações.

3.6 Modelo Entidade Relacionamento

Figura 17 – Modelo Entidade Relacionamento do aplicativo



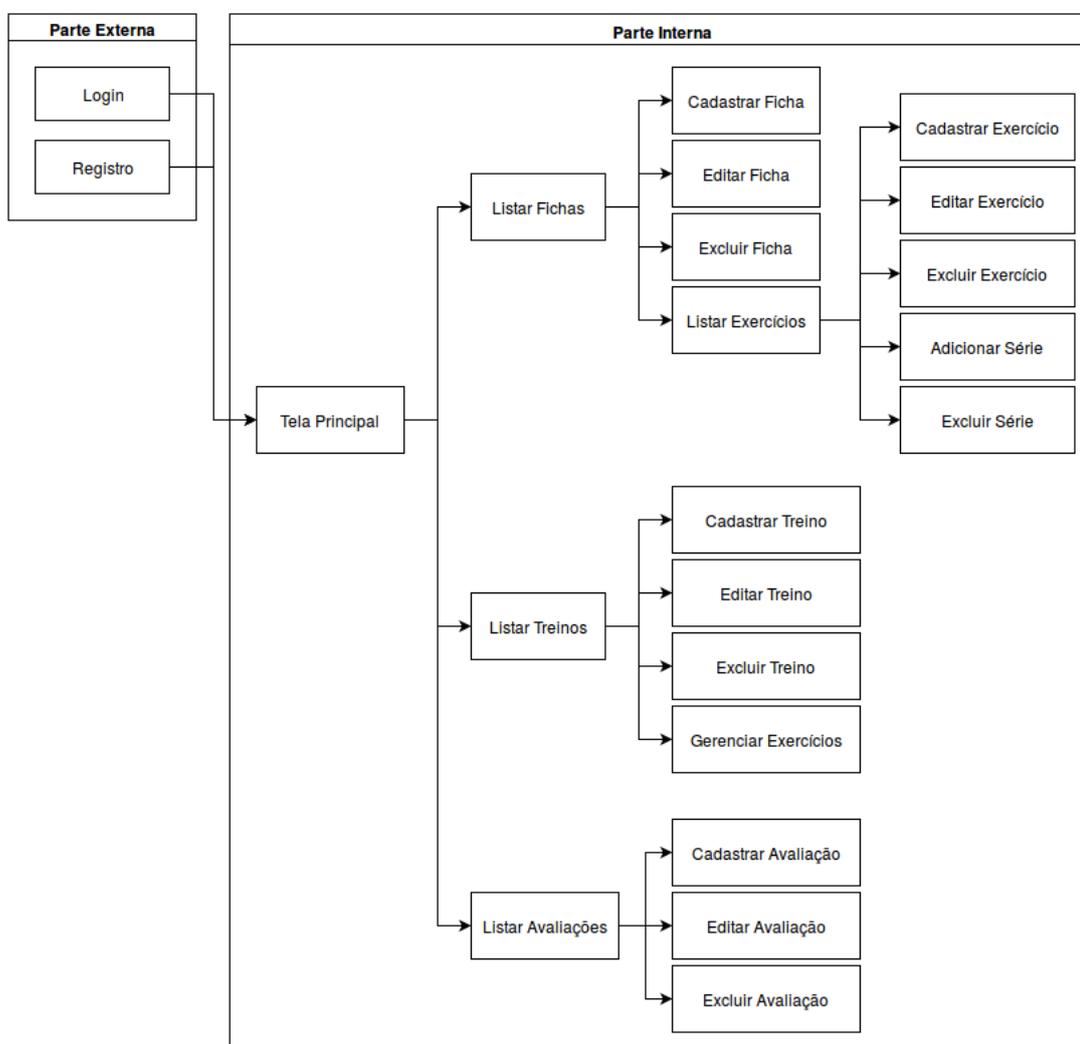
Fonte: Elaborado pelo autor

De acordo com os dados levantados nos requisitos, foi criado um modelo entidade relacionamento (MER). Esse modelo, utilizado na Engenharia de Software e em Banco de Dados, é responsável por representar as entidades do sistema, bem como seus atributos e os relacionamentos entre essas entidades. Através do MER, conseguimos ter uma visão melhor de como os objetos se relacionam dentro do sistema de banco de dados.

A Figura 17 ilustra o modelo entidade relacionamento criado para o aplicativo desse projeto. O Anexo B apresenta uma estrutura descritiva do banco de dados, o que ajuda a visualizar de forma mais técnica o esquema ilustrado no DER.

3.7 Diagrama de funcionalidades

Figura 18 – Diagrama de funcionalidades do aplicativo



Fonte: Elaborado pelo autor

Na Figura 18, temos o diagrama de funcionalidades do aplicativo. Essas funcionalidades estão de acordo com todas as especificações no documento de requisitos, descrito na seção 4.3.

Como podemos perceber na Figura 18, todas as funcionalidades estão organizadas de forma hierárquica e as setas representam o fluxo de acesso. Por exemplo, através da funcionalidade de Listar Treinos, podemos ter acesso às funcionalidades de Cadastrar Treino, Editar Treino, Excluir Treino, Gerenciar Exercícios de um determinado treino.

O sistema é composto por duas partes: parte externa e interna. Na parte externa, temos as funcionalidades de criação e acesso à conta. O usuário consegue ter acesso às funcionalidades internas somente por essas duas alternativas. Na parte interna, todos os módulos principais de Ficha, Treino e Avaliação são acessados pela tela principal. Por

serem módulos distintos, cada um apresenta uma tela separada. Dentro do módulo de Fichas, temos outros submódulos que fazem parte do módulo em questão, e podem ser acessados apenas dentro dele.

4 Metodologia

Este capítulo mostra como o trabalho foi realizado, qual o ambiente e as tecnologias utilizadas (seção 4.1), como foi estruturado o sistema (seção 4.2), quais documentos foram gerados (seções 4.3, 3.6 e 3.7) e as etapas do desenvolvimento (seção 4.4).

4.1 Ambiente de desenvolvimento

O aplicativo foi desenvolvido em um *notebook* pessoal, com as seguintes configurações de *hardware* e *software*:

- **Processador:** Intel® Core™ i5-3210M CPU @ 2.50GHz x 2.
- **Tipo de Sistema Operacional:** 64-bit.
- **Memória RAM:** 3,6 GiB.
- **Disco rígido:** 500GB.
- **Sistema Operacional:** Ubuntu 16.04.1 LTS (Xenial Xerus).

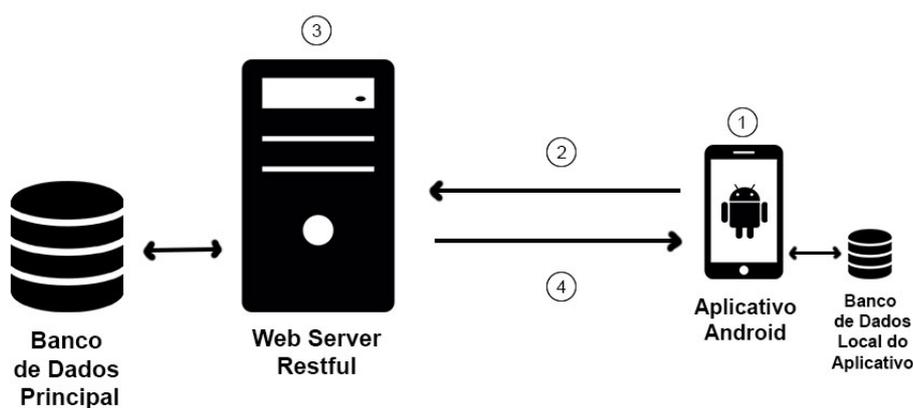
Além disso, foi necessário preparar o ambiente de desenvolvimento, que conta com alguns *softwares* voltados para a criação desse tipo de aplicação. Os *softwares* utilizados foram:

- **JDK (versão 8):** O JDK é um ambiente de desenvolvimento para a criação de aplicativos, *applets*, e componentes usando a linguagem de programação Java.
- **Tomcat (versão 8.0.27):** O Tomcat é um *software* de servidor *web* de código fonte aberto responsável por implementar as tecnologias Java Servlet, JavaServer Pages, Java Expression Language e Java WebSocket.
- **Eclipse (distribuição Mars, versão 4.5.0):** O Eclipse é um ambiente de desenvolvimento integrado usado para programação em diversas linguagens, principalmente Java. Esse ambiente conta com um sistema de *plug-ins* que podem ser usados para customizar o ambiente de acordo com a linguagem de programação e o tipo de aplicação a ser desenvolvida.
- **Android Studio (versão 2.1.3):** O Android Studio é um ambiente de desenvolvimento integrado oficial da Google para criar aplicações Android. Ele conta com diversos componentes que dão suporte ao desenvolvimento de aplicativos, tanto para *smartphones* e *tablets*, quanto para Android Wear e outras plataformas.
- **Git (versão 2.7.4) e GitHub:** O Git é um sistema de controle de versão para desenvolvimento de *softwares*. Ele tem como características principais a velocidade, integridade dos dados e suporte para fluxos de trabalho distribuídos e não-lineares. O GitHub (GitHub, Inc., 2016), por sua vez, é um sistema *online* para hospedagem de

repositórios Git. Esse sistema é muito popular entre desenvolvedores que pretendem disponibilizar o código fonte de seus projetos na internet para que outras pessoas possam colaborar.

4.2 Estrutura do sistema

Figura 19 – Representação ilustrativa da estrutura de funcionamento do sistema.



Fonte: Elaborado pelo autor

O sistema foi projetado para ter uma estrutura bem simples. Esse sistema é composto pela arquitetura Cliente-Servidor, onde existe um servidor central e vários clientes que se conectam a ele. A Figura 19 faz uma ilustração do fluxo do sistema, descrita a seguir:

A Figura 19(1) representa o aplicativo, sendo executado em um dispositivo móvel que tem como base a plataforma Android. O aplicativo foi desenvolvido utilizando a linguagem Java, e ele conta com um banco de dados local em SQLite.

Para gerenciar o banco de dados local do aplicativo, foi utilizada uma biblioteca personalizada chamada Realm (REALM.IO, 2016). O Realm é uma biblioteca ORM para Android que serve como uma alternativa ao SQLite padrão. Essa biblioteca conta com um banco de dados orientado a objetos que garante rapidez e facilidade de uso na manipulação dos registros e realização das consultas.

Para gerenciar as requisições enviadas do aplicativo ao servidor central, foi utilizada uma biblioteca personalizada chamada Retrofit (SQUARE, 2016). O Retrofit é uma biblioteca responsável por gerenciar requisições RESTful em aplicativos Android. Essa biblioteca conta com uma série de classes e anotações que auxiliam na manipulação das requisições

realizadas ao servidor, além de dar suporte a monitoramento dos *logs* das requisições enviadas e suas respostas.

O aplicativo contém todas as funcionalidades que poderão ser utilizadas pelo usuário. Todas as alterações realizadas pelo usuário são armazenadas localmente primeiro. Quando houver conexão com a internet, o aplicativo faz requisições ao servidor para sincronizar os dados.

A Figura 19(2) representa uma requisição RESTful realizada ao servidor. Sempre que o usuário estiver conectado, o aplicativo faz requisições ao servidor enviando os dados locais do aplicativo, e quais alterações foram feitas nesses dados para que estas possam ser replicadas ao banco de dados principal.

A Figura 19(3) representa o servidor, sendo executado em um computador *desktop* comum, e que pode ser acessado por um IP e porta fixos. O servidor também conta com um banco de dados principal, onde ficam armazenados os dados de todos os usuários.

Para gerenciar as requisições RESTful recebidas pelo *container* do servidor, foi utilizado o *framework* Spring. Com esse *framework* foi possível criar uma interface única para as requisições trocadas entre aplicativo e servidor, além de manipular diversas rotas para diferentes requisições e oferecer serviços para cada operação específica.

Para gerenciar a parte do banco de dados, foi utilizado o *framework* Hibernate. Esse *framework* é uma ferramenta ORM que permite criar objetos relacionais e mapeá-los de acordo com a estrutura do banco de dados, além de oferecer funções para manipulação e consulta dos registros.

A Figura 19(4) Esse item representa uma resposta do servidor ao aplicativo que fez a requisição. A resposta contém os resultados das operações que foram realizadas no servidor, permitindo o aplicativo identificar se houve sucesso ou falha. Além disso, a resposta também contém os dados atuais do servidor, que serão replicados para o aplicativo, com isso fazendo a sincronização.

Com essa estrutura, foi possível criar um esquema de comunicação entre o servidor e os aplicativos que rodam em diversos dispositivos Android. Além disso, foi possível oferecer uma ferramenta com funcionamento *offline* (seção 3.3) e disponibilidade de dados (seção 3.4).

4.3 Documento de requisitos

O documento de requisitos é um documento de grande importância no processo de desenvolvimento de um *software*. Esse documento é responsável por delimitar o escopo do projeto, além de definir quais serão as funcionalidades e os atributos de segurança, desempenho, dentre outros fatores de qualidade.

Para esse projeto, foi criado um exemplo de documento de requisitos, que encontra-se no Anexo A. Esse documento contém uma descrição simples de cada requisito do sistema, e também contém os casos de uso de cada funcionalidade.

O documento de requisitos foi construído através de algumas pesquisas nas academias, onde foram realizadas conversas informais. Com isso, foi possível levantar quais são os principais problemas vivenciados pelas pessoas naquele ambiente e quais são suas principais necessidades. Depois disso, essas informações foram refinadas e transformadas em requisitos. Estes requisitos foram avaliados para saber quais seriam viáveis de serem implementados na aplicação a ser desenvolvida.

4.4 Controle de Versionamento e Configuração

Para manter um nível de qualidade e organizar o processo de desenvolvimento de *software* é necessário fazer o controle de versionamento e configuração do sistema. Essa técnica consiste em registrar o estado atual do sistema em várias épocas do desenvolvimento para que se mantenha um histórico de alterações realizadas. Também podemos recuperar arquivos que foram danificados ou modificados de forma indevida nas versões atuais, sem comprometer o andamento do projeto.

Uma característica essencial de qualquer ferramenta de controle de versionamento é a capacidade de juntar arquivos que foram modificados por duas ou mais fontes diferentes, oferecendo formas de resolver os conflitos. Essa característica é mais indicada para projetos em que duas ou mais pessoas trabalham em conjunto, e frequentemente fazem alterações nos mesmos arquivos.

O *software* utilizado neste projeto para fazer o controle de versionamento foi o Git devido a sua robustez, popularidade e facilidade de uso, de acordo com as métricas do Google Trends (Google Trends, 2016).

Quadro 1 – Controle de versionamento e configuração do aplicativo desenvolvido.

Versão	Data	Descrição
0.5	17/08	Preparação do ambiente de desenvolvimento. Criação do diagrama de entidades do banco de dados. Criação do <i>webserver</i> na ferramenta Eclipse. Criação do aplicativo na ferramenta Android Studio.
1.0	02/09	Desenvolvimento da funcionalidade <i>login</i> e registro de usuários.
2.0	23/09	Desenvolvimento da funcionalidade ficha de exercícios. Correções nas funcionalidades da versão anterior. Alterações no banco de dados e no diagrama.
3.0	14/10	Desenvolvimento da funcionalidade treinos. Correções nas funcionalidades da versão anterior. Alterações no banco de dados e no diagrama.
4.0	28/10	Desenvolvimento da funcionalidade avaliações físicas. Correções nas funcionalidades da versão anterior. Alterações no banco de dados e no diagrama.
4.5	11/11	Correções gerais e fechamento do sistema.

Um repositório local foi criado para armazenar os arquivos do projeto. O desenvolvimento foi dividido em algumas etapas distintas, cada uma representando um módulo diferente. O Quadro 1 representa como foi feito o controle de versão do projeto de desenvolvimento do aplicativo.

Nesse quadro, a coluna Versão representa uma espécie de *tag* identificadora para cada versão fechada do sistema. A coluna Data representa qual o dia exato ocorreu o fechamento da versão no repositório. A coluna Descrição abrange as características de cada versão, apontando o que foi modificado e quais funcionalidades foram desenvolvidas.

5 Conclusões

A popularização dos dispositivos móveis nos dias atuais possibilitou o surgimento da demanda por diversas aplicações que facilitam a vida das pessoas oferecendo ferramentas práticas que podem ser usadas nas tarefas do dia a dia.

Este trabalho teve como objetivo a criação de um aplicativo que possa auxiliar no gerenciamento de fichas de exercícios, gerenciamento de treinos e avaliações físicas dos alunos no ambiente das academias. Além disso, foi proposto o estudo das tendências atuais na área de desenvolvimento de aplicativos para Android, levando em consideração a abordagem de diversas técnicas. Podemos citar, por exemplo, a arquitetura cliente-servidor utilizando a troca de requisições RESTful como forma de comunicação, o uso do SQLite para fazer o armazenamento de dados *offline* na base local do aplicativo, e também o uso de bibliotecas como Retrofit e *frameworks* como Spring e Hibernate que auxiliam no desenvolvimento tornando o código fonte mais bem estruturado, aumentando sua legibilidade e a robustez.

Como conclusões, podemos dizer que foi possível criar a ferramenta proposta utilizando as tecnologias atuais para desenvolvimento de aplicativos. O objetivo principal de estudar as tecnologias mais recentes e desenvolver um produto baseado nessas tecnologias e nas práticas de engenharia de *software* foi alcançado.

De acordo com os resultados, foi possível perceber que o aplicativo oferece recursos básicos para atender a demanda dos processos de treinos e avaliações realizados pelos alunos na academia. Podemos dizer que o produto desenvolvido é viável, o aplicativo tem a capacidade de atender a demanda para a qual ele foi criado, além de se enquadrar bem no padrão dos aplicativos do mesmo ramo já presentes no mercado. Além disso, ele conta com alguns recursos diferenciais de funcionamento *offline* e disponibilidade de dados.

Como trabalhos futuros, é possível fazer a aplicação dessa ferramenta em academias reais, para conseguir medir o nível de satisfação dos usuários. Para isso, será necessário hospedar o aplicativo na Play Store.

Também é vislumbrada a possibilidade de criar novos recursos dentro do aplicativo para permitir que haja uma espécie de perfil para professores, alunos e avaliadores físicos. E, juntamente com essa idéia, elaborar formas de interação entre esses três perfis, incentivando a interação social entre esses diferentes tipos de pessoa.

Outra incrementação que pode ser feita nesse *software* é a criação de novos módulos relacionados à área abordada, como por exemplo um módulo para gerenciar dietas, um módulo para calculadoras de necessidade calórica, dentre outros.

Referências

Adobe Systems Inc. **PhoneGap Official Website**. 2016. Disponível em: <<http://phonegap.com/>>. Acesso em: 10 de outubro de 2016. Citado na página 10.

ALECRIM, E. **O que é Tecnologia da Informação (TI)?** 2015. Disponível em: <<http://www.infowester.com/ti.php>>. Acesso em: 10 de novembro de 2016. Citado na página 4.

AMADOR, J. G. **Estudo mostra crescimento no uso de dispositivos móveis e domínio Android**. 2015. Disponível em: <http://www.correiobrasiliense.com.br/app/noticia/tecnologia/2015/01/17/interna_tecnologia,466691/estudo-mostra-crescimento-no-uso-de-dispositivos-moveis-e-dominio-andr.shtml>. Acesso em: 25 de novembro de 2016. Citado na página 2.

apache.org. **Apache HTTP Server Project**. 2016. Disponível em: <<https://httpd.apache.org/>>. Acesso em: 22 de outubro de 2016. Citado na página 11.

App Store. **Site Oficial da App Store**. 2016. Disponível em: <<https://itunes.apple.com/us/genre/ios/id36?mt=8>>. Acesso em: 19 de outubro de 2016. Citado na página 8.

APPLE. **Site Oficial do iOS 9**. 2016. Disponível em: <<http://www.apple.com/ios/>>. Acesso em: 12 de julho de 2016. Citado na página 7.

BENTO, A. R. et al. Importância da tecnologia no acompanhamento das atividades dos alunos de academia. **Convibra**, 2013. Citado na página 15.

DB-ENGINES. **DB-Engines Ranking - historical trend of the popularity ranking of database management systems**. 2016. Disponível em: <http://db-engines.com/en/ranking_trend>. Acesso em: 23 de julho de 2016. Citado na página 13.

DB-ENGINES. **DB-Engines Ranking - popularity ranking of database management systems**. 2016. Disponível em: <<http://db-engines.com/en/ranking>>. Acesso em: 23 de julho de 2016. Citado na página 13.

FACEBOOK. **Site Oficial do Facebook**. 2016. Disponível em: <<https://www.facebook.com>>. Acesso em: 13 de julho de 2016. Citado na página 9.

G1. **Uso de smartphones e redes sociais vira febre nas academias**. 2013. Disponível em: <<http://g1.globo.com/rs/rio-grande-do-sul/jornal-do-almoco/videos/v/uso-de-smartphones-e-redes-sociais-vira-febre-nas-academias/2942522/>>. Acesso em: 29 de novembro de 2016. Citado na página 1.

GARTNER. **Gartner Says Worldwide Smartphone Sales Grew 9.7 Percent in Fourth Quarter of 2015**. 2016. Disponível em: <<http://www.gartner.com/newsroom/id/3215217>>. Acesso em: 27 de maio de 2016. Citado na página 7.

GitHub, Inc. **Site Oficial do GitHub**. 2016. Disponível em: <<https://github.com/>>. Acesso em: 26 de novembro de 2016. Citado na página 31.

GNU. **GNU General Public License v3.0**. 2016. Disponível em: <<https://www.gnu.org/licenses/gpl-3.0.en.html>>. Acesso em: 14 de julho de 2016. Citado na página 9.

GOOGLE. **Site Oficial do Google Android**. 2016. Disponível em: <<https://www.android.com/>>. Acesso em: 12 de julho de 2016. Citado na página 6.

Google Trends. **Comparação do nível de popularidade entre as ferramentas de controle de versionamento**. 2016. Disponível em: <https://www.google.com/trends/explore?date=all&q=%2Fm%2F05vqwg,%2Fm%2F012ct9,%2Fm%2F08441_,%2Fm%2F08w6d6,%2Fm%2F09d6g&hl=en-US>. Acesso em: 20 de outubro de 2016. Citado na página 34.

GOWIN, M. et al. Health and fitness app use in college students: A qualitative study. **American Journal of Health Education**, v. 1, p. 223–230, 2014. Citado na página 15.

GROSSMANN, L. O. **Mundo tem 7,1 bilhões de celulares ativos**. 2015. Disponível em: <<http://convergenciadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=site&inoid=40220&sid=8>>. Acesso em: 19 de maio de 2016. Citado na página 5.

GUEDES, J. G. F. **Uma Proposta de Solução para Funcionamento Offline em Aplicações Android**. Tese (Trabalho de Conclusão do Curso Mestrado Acadêmico em Sistemas e Computação) — Universidade Federal do Rio Grande do Norte, Natal-RN, 2015. Citado na página 23.

hibernate.org. **Hibernate Official Website**. 2016. Disponível em: <<http://hibernate.org/>>. Acesso em: 28 de outubro de 2016. Citado na página 12.

IDC. **Smartphone OS Market Share, 2015 Q2**. 2015. Disponível em: <<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>>. Acesso em: 27 de maio de 2016. Citado na página 7.

IDC. **Smartphone Vendor Market Share, 2015 Q2**. 2016. Disponível em: <<http://www.idc.com/prodserv/smartphone-market-share.jsp>>. Acesso em: 19 de julho de 2016. Citado na página 1.

INSTAGRAM. **Site Oficial do Instagram**. 2016. Disponível em: <<https://www.instagram.com>>. Acesso em: 13 de julho de 2016. Citado na página 9.

IT Forum 365. **Demanda por desenvolvimento de aplicativos corporativos irá crescer até 2017**. 2015. Disponível em: <<http://www.itforum365.com.br/noticias/detalhe/116084/demanda-por-desenvolvimento-de-aplicativos-corporativos-ira-crescer-ate-2017>>. Acesso em: 10 de novembro de 2016. Citado na página 2.

JSON.ORG. **Introducing JSON**. 2016. Disponível em: <<http://www.json.org/>>. Acesso em: 19 de julho de 2016. Citado na página 11.

KALALI, M.; MEHTA, B. **Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON**. [S.l.]: Packt Publishing Ltd., 2013. Citado na página 14.

LECHETA, R. R. **Google Android**. São Paulo: Novatec, 2013. Citado na página 5.

LEE, V.; SCHNEIDER, H.; SCHELL, R. **Aplicações móveis: arquitetura, projeto e desenvolvimento**. São Paulo: Pearson, 2005. Citado na página 5.

linux.com. **What is Linux?** 2015. Disponível em: <<https://www.linux.com/what-is-linux>>. Acesso em: 15 de novembro de 2016. Citado na página 6.

LITMAN, L. et al. Mobile exercise apps and increased leisure time exercise activity: A moderated mediation analysis of the role of self-efficacy and barriers. **Journal of Medical Internet Research**, 2015. Citado na página 16.

MICROSOFT. **The Official Microsoft IIS Site**. 2016. Disponível em: <<https://www.iis.net/>>. Acesso em: 22 de outubro de 2016. Citado na página 11.

MICROSOFT. **Site Oficial do Microsoft Windows Phone**. 2016. Disponível em: <<https://www.microsoft.com/pt-br/windows/phones>>. Acesso em: 12 de julho de 2016. Citado na página 8.

Microsoft Corporation. **About Xamarin**. 2016. Disponível em: <<https://www.xamarin.com/about>>. Acesso em: 10 de outubro de 2016. Citado na página 11.

Microsoft Corporation. **Xamarin Official Website**. 2016. Disponível em: <<https://www.xamarin.com/>>. Acesso em: 10 de outubro de 2016. Citado na página 11.

Mozilla Developer Network. **AJAX: Primeiros Passos**. 2016. Disponível em: <https://developer.mozilla.org/pt-BR/docs/AJAX/Getting_Started>. Acesso em: 22 de outubro de 2016. Citado na página 12.

NGINX Inc. **NGINX Official Website**. 2016. Disponível em: <<https://www.nginx.com/>>. Acesso em: 22 de outubro de 2016. Citado na página 11.

ORACLE. **Site Oficial do Oracle Java**. 2016. Disponível em: <<https://www.java.com>>. Acesso em: 14 de julho de 2016. Citado na página 9.

Oracle Corporation. **MySQL Official Website**. 2016. Disponível em: <<https://www.mysql.com/>>. Acesso em: 19 de novembro de 2016. Citado na página 12.

PARIS, L. R. B.; SANTOS, T. Y. V. dos; JUNIOR, V. A. **FITNESS APP: Cuidando da sua saúde**. Tese (Trabalho de conclusão de curso) — Centro Estadual De Educação Tecnológica Paula Souza, Palmital, 2015. Citado na página 15.

Pivotal Software. **Site oficial do Spring Framework**. 2016. Disponível em: <<https://projects.spring.io/spring-framework/>>. Acesso em: 23 de julho de 2016. Citado na página 12.

REALM.IO. **Realm for Android**. 2016. Disponível em: <<https://realm.io/news/realm-for-android/>>. Acesso em: 28 de setembro de 2016. Citado na página 32.

ROUSE, M. **Definition of HTTP (Hypertext Transfer Protocol)**. 2006. Disponível em: <<http://searchwindevelopment.techtarget.com/definition/HTTP>>. Acesso em: 20 de outubro de 2016. Citado na página 14.

ROUSE, M. **What is Web Server?** 2015. Disponível em: <<http://whatis.techtarget.com/definition/Web-server>>. Acesso em: 20 de outubro de 2016. Citado na página 11.

ROUSE, M. **Definition of SQL (Structured Query Language)**. 2016. Disponível em: <<http://searchsqlserver.techtarget.com/definition/SQL>>. Acesso em: 10 de novembro de 2016. Citado na página 15.

SQUARE. **Retrofit**. 2016. Disponível em: <<http://square.github.io/retrofit/>>. Acesso em: 28 de setembro de 2016. Citado na página 32.

TIOBE. **TIOBE Index for May 2016**. 2016. Disponível em: <http://www.tiobe.com/tiobe_index>. Acesso em: 17 de maio de 2016. Citado na página 10.

TOPOL, E. J. **The Creative Destruction of Medicine: How the Digital Revolution Will Create Better Health Care**. Revised and expanded. [S.l.]: Basic Books; First Trade Paper Edition, 2013. Citado na página 15.

TWITTER. **Site Oficial do Twitter**. 2016. Disponível em: <<https://www.twitter.com>>. Acesso em: 13 de julho de 2016. Citado na página 9.

VASILE, C. **Android 5.0 Lollipop Developer Preview - Screenshot Tour**. 2014. Disponível em: <<http://news.softpedia.com/news/Android-5-0-Lollipop-Developer-Preview-Screenshot-Tour-462596.shtml>>. Acesso em: 12 de julho de 2016. Citado na página 6.

VASILE, C. **Windows 10 for Phones Preview Screenshots Gallery**. 2015. Disponível em: <<http://news.softpedia.com/news/Windows-10-for-Phones-Preview-Screenshots-Gallery-473009.shtml>>. Acesso em: 12 de julho de 2016. Citado na página 9.

VILLAS-BOAS, A. **9 cool features coming to your iPhone and iPad that Apple didn't mention yet**. 2015. Disponível em: <<http://uk.businessinsider.com/8-cool-features-coming-to-your-iphone-and-ipad-that-apple-didnt-mention-yet-2015-6>>. Acesso em: 12 de julho de 2016. Citado na página 8.

W3TECHS. **Usage of web servers for websites**. 2016. Disponível em: <https://w3techs.com/technologies/overview/web_server/all>. Acesso em: 27 de maio de 2016. Citado na página 11.

WALSH, N. **A Technical Introduction to XML - What is XML?** 2016. Disponível em: <<http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58>>. Acesso em: 19 de julho de 2016. Citado na página 11.

Anexos

ANEXO A – Documento de Requisitos

Marco Aurélio

Documento de Requisitos do Aplicativo para Academia

Sabará

2016, v1.0

Sumário

1	INTRODUÇÃO	2
2	REQUISITOS FUNCIONAIS	3
2.1	Dados da Conta	3
2.1.1	RF001-1 - Criar conta	3
2.1.2	RF001-2 - Acessar conta	3
2.2	Ficha de Exercícios	4
2.2.1	RF002-1 - Listar fichas	5
2.2.2	RF002-2 - Cadastrar fichas	5
2.2.3	RF002-3 - Editar fichas	5
2.2.3.1	RF002-3a - Cadastrar exercícios	6
2.2.3.2	RF002-3b - Editar exercícios	6
2.2.3.3	RF002-3c - Excluir exercícios	6
2.2.4	RF002-4 - Excluir fichas	6
2.3	Treinos	6
2.3.1	RF003-1 - Listar treinos	7
2.3.2	RF003-2 - Cadastrar treinos	7
2.3.3	RF003-3 - Editar treinos	7
2.3.4	RF003-4 - Excluir treinos	7
2.4	Avaliações Físicas	8
2.4.1	RF004-1 - Listar avaliações físicas	8
2.4.2	RF004-2 - Cadastrar avaliações físicas	8
2.4.3	RF004-3 - Editar avaliações físicas	9
2.4.4	RF004-4 - Excluir avaliações físicas	9
3	REQUISITOS NÃO FUNCIONAIS	10
3.1	Usabilidade	10
3.2	Navegação	10
3.3	Privacidade	10
3.4	Desempenho	10
	REFERÊNCIAS	11

1 Introdução

Este documento tem como finalidade fazer uma apresentação detalhada dos requisitos levantados na criação do aplicativo para academia. Esse aplicativo conta com funcionalidades que auxiliam as pessoas da academia nas suas atividades.

O documento apresenta dois capítulos principais. Requisitos Funcionais e Requisitos Não Funcionais.

Nos Requisitos Funcionais (capítulo 2) são apresentados os principais módulos do sistema, como eles devem se comportar, e quais as funções podem ser realizadas.

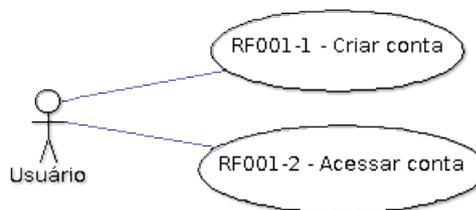
Nos Requisitos Não Funcionais (capítulo 3) são apresentados os requisitos de implementação, tais como segurança, desempenho, usabilidade, dentre outros.

2 Requisitos Funcionais

2.1 Dados da Conta

Essa seção descreve os requisitos de dados da conta. A Figura 1 representa o diagrama de caso de uso para os requisitos dessa seção.

Figura 1 – Diagrama de caso de uso Dados da Conta



Fonte: Elaborado pelo autor

2.1.1 RF001-1 - Criar conta

O usuário pode criar uma nova conta no aplicativo. Deve ser disponibilizado um formulário com os campos Nome, Email e Senha. Os dados fornecidos pelo usuário devem ser avaliados, e se tudo estiver de acordo, uma nova conta é criada e o usuário pode ter acesso aos módulos internos do aplicativo.

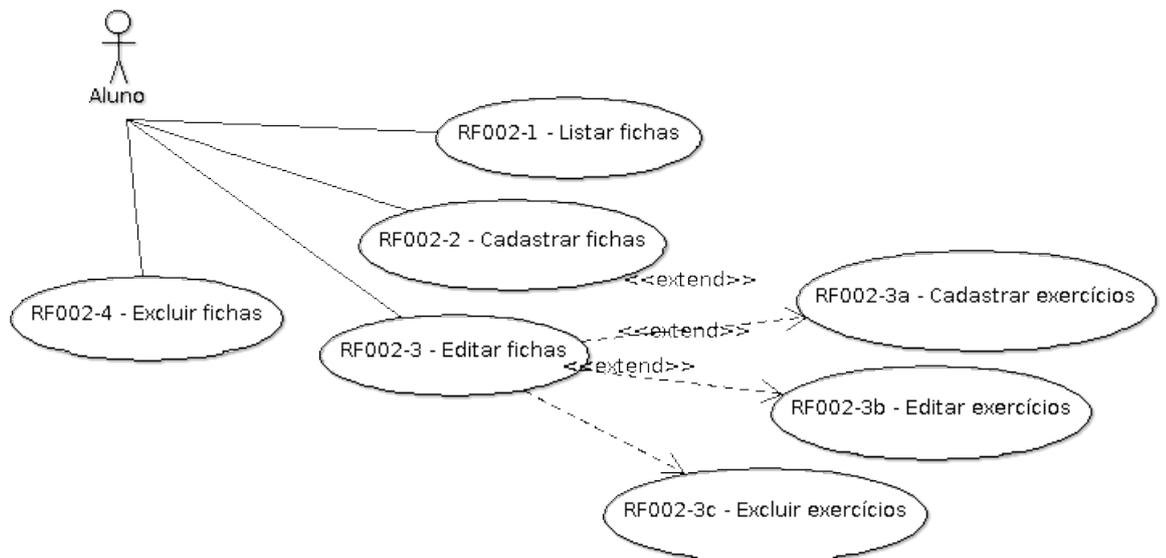
2.1.2 RF001-2 - Acessar conta

O usuário pode acessar sua conta no aplicativo. Deve ser disponibilizado um formulário com os campos Email e Senha. O sistema deve verificar a autenticidade dos dados, e se tudo estiver correto, o usuário é redirecionado para a parte interna do sistema.

2.2 Ficha de Exercícios

Essa seção descreve os requisitos do módulo de ficha de exercícios. A Figura 2 representa o diagrama de caso de uso para os requisitos dessa seção.

Figura 2 – Diagrama de caso de uso do módulo Ficha de Exercícios



Fonte: Elaborado pelo autor

A ficha de exercícios consiste em um artefato que auxilia o aluno a se organizar durante a execução dos exercícios. Ela apresenta uma listagem dos exercícios, com seus determinados pesos e séries, na ordem em que devem ser executados. A ficha também costuma apresentar algumas observações para que o aluno lembre-se de alguns detalhes. Ela serve como um instrumento de consulta, serve também como uma maneira de o professor acompanhar o treino de seus alunos.

A Figura 3 ilustra um exemplo de um plano de treino de uma academia fictícia. Nesse plano de treino, há duas fichas registradas para um aluno.

Figura 3 – Exemplo fictício de um plano de treinos para academia.

Academia XYZ - Plano de treino - Data: 10/07/2012			
Aluno: Fulano			
Ficha A			
Dias: Segunda e Quinta			
Exercícios	Séries	Repetições	Carga
Agachamento fundo	3	10	90Kg
Flexão de braços	3	12	40Kg
Abdominal crunch	3	10	20Kg
Tríceps testa	3	10	30Kg
Ficha B			
Dias: Terça e Sexta			
Exercícios	Séries	Repetições	Carga
Leg 45°	3	10	140Kg
Extensão dorsal	3	10	20Kg
Puxada pulley pela frente	3	10	45Kg
Remada curvado	3	10	35Kg

Fonte: Elaborado pelo autor

2.2.1 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.

2.2.2 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.

2.2.3 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.

2.2.3.1 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.

2.2.3.2 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.

2.2.3.3 RF002-3c - Excluir exercícios

O aluno pode excluir um exercício da ficha.

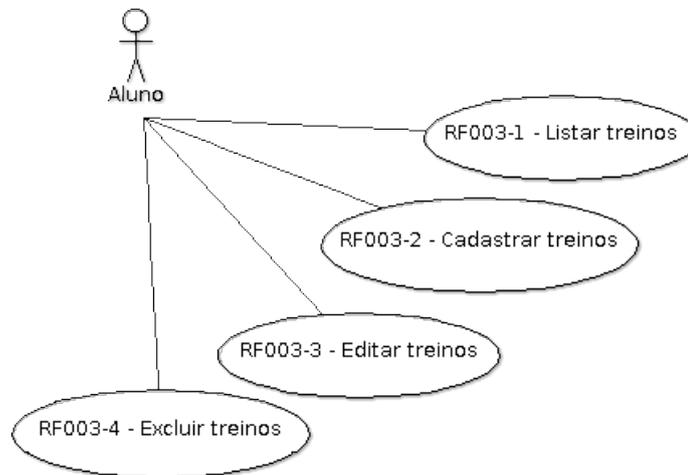
2.2.4 RF002-4 - Excluir fichas

O aluno pode excluir uma ficha de exercício cadastrada.

2.3 Treinos

Essa seção descreve os requisitos do módulo de treinos. A Figura 4 representa o diagrama de caso de uso para os requisitos dessa seção. Os treinos funcionam como uma espécie de histórico que mostra qual o dia e o horário que o aluno realizou um treino e quais exercícios foram feitos naquele treino.

Figura 4 – Diagrama de caso de uso do módulo Treinos



Fonte: Elaborado pelo autor

2.3.1 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.

2.3.2 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.

2.3.3 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.

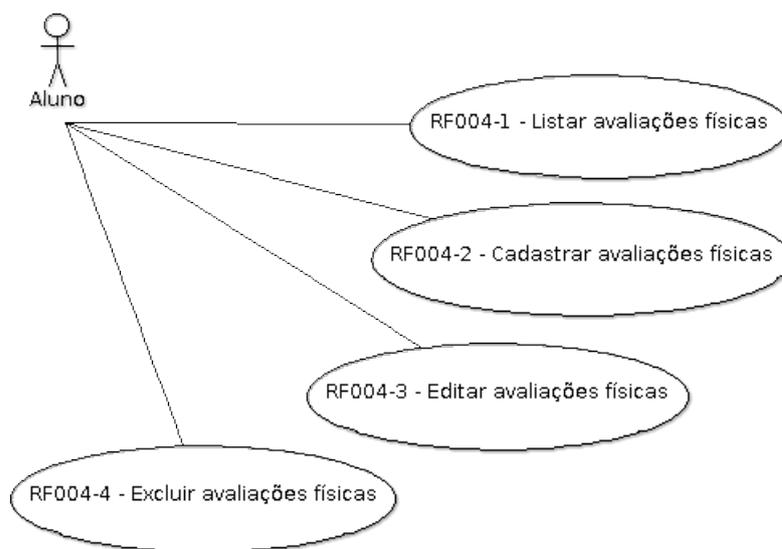
2.3.4 RF003-4 - Excluir treinos

O aluno pode excluir um treino cadastrado.

2.4 Avaliações Físicas

Essa seção descreve os requisitos do módulo de avaliações físicas. A Figura 5 representa o diagrama de caso de uso para os requisitos dessa seção.

Figura 5 – Diagrama de caso de uso do módulo Avaliações Físicas



Fonte: Elaborado pelo autor

2.4.1 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.

2.4.2 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). A Figura 6 mostra quais são os campos que deverão conter no formulário.

Figura 6 – Tela de avaliação físicas do *software* Corpore Teste

Corpore Teste - Cadastro de Alunos

Cadastro: Consulta: Alterações: Resultados de avaliação: Resultados morfológicos:

Resultados anamnese: Medidas Fisiológicas: Medidas Antropométricas

Medidas:

Selecione o aluno: Data:

Dobras Cutâneas:

Peitoral: mm

Bíceps: mm

Tríceps: mm

Subescapular: mm

Supra-iliaca: mm

Axilar-média: mm

Abdominal: mm

Cova: mm

Panturrilha medial: mm

Medidas Lineares (circunferências)

Tórax: cm

Abdômen: cm

Cintura: cm

Quadril: cm

Braço direito: cm

Braço esquerdo: cm

Antebraço direito: cm

Antebraço esquerdo: cm

Cova direita: Proximal: Medial: Distal: cm cm cm

Medidas Transversais (diâmetros):

Biestilóide (punho): cm

Biepicondiliano do fêmur: cm

Biacromial: cm

Bileocristal: cm

Bitrocantariano: cm

Bimaleolar: cm

Biepicondiliano do úmero: cm

Cova esquerda: Proximal: Medial: Distal: cm cm cm

Perna direita: cm

Perna esquerda: cm

Ombros: cm

Pescoço: cm

Punho: cm

Joelho: cm

Tornozelo: cm

Imprimir < Voltar Sair

Fonte: (Corpore Teste, 2016)

2.4.3 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.

2.4.4 RF004-4 - Excluir avaliações físicas

O aluno pode excluir uma avaliação física cadastrada.

3 Requisitos Não Funcionais

3.1 Usabilidade

O sistema deve contar com um *layout* simples e intuitivo. As telas devem conter o mesmo padrão de cor, estilização de botões e tamanhos de fontes.

Os ícones devem ser fáceis de compreender e assimilar com suas respectivas funções, para que não haja a necessidade de colocar textos explicativos.

Os formulários de edição precisam vir preenchidos com os dados atuais do registro, para que o usuário não precise digitar todos os dados novamente antes de alterá-los.

Em todas as funcionalidades que envolverem exclusão de registro, deve abrir um diálogo perguntando se o usuário realmente pretende realizar aquela ação de risco.

3.2 Navegação

O sistema deve oferecer uma navegação simples, capaz de proporcionar o usuário uma facilidade maior na localização dos elementos que ele procura.

A barra superior do aplicativo deve sempre informar em qual módulo o usuário está presente, para que torne a orientação mais fácil.

3.3 Privacidade

Cada usuário possui uma conta distinta, porém nenhum usuário pode ter acesso a informações de outros usuários. O sistema deve tratar as requisições de acordo com o identificador de cada usuário.

3.4 Desempenho

O sistema deve ser capaz de realizar as funções de forma rápida, de forma que o usuário não sinta nenhum atraso no processamento, para não causar perda de tempo a este usuário.

Em casos que podem ocorrer atrasos, o sistema deve informar anteriormente, para que o usuário fique ciente de que naquela determinada operação ele deve esperar um pouco mais.

Referências

Corpore Teste. *Corpore Teste - Avaliação Física*. 2016. Disponível em: <<http://www.corporeteste.com.br/site/>>. Acesso em: 20 de agosto de 2016. Citado 2 vezes nas páginas 8 e 9.

ANEXO B – Estrutura Descritiva do Banco de Dados

avaliacao

Column	Type	Null	Default	Comments
id (Primary)	int(11)	No		
altura	float	No		
peso	float	No		
peitoral	float	Yes	NULL	
biceps	float	Yes	NULL	
triceps	float	Yes	NULL	
subescapular	float	Yes	NULL	
supra_iliaca	float	Yes	NULL	
axiliar_media	float	Yes	NULL	
abdominal	float	Yes	NULL	
coxa	float	Yes	NULL	
panturrilha_media	float	Yes	NULL	
torax	float	Yes	NULL	
abdomen	float	Yes	NULL	
cintura	float	Yes	NULL	
quadril	float	Yes	NULL	
braco_direito	float	Yes	NULL	
braco_esquerdo	float	Yes	NULL	
antebraco_direito	float	Yes	NULL	
antebraco_esquerdo	float	Yes	NULL	
coxa_direita	float	Yes	NULL	
coxa_esquerda	float	Yes	NULL	
perna_direita	float	Yes	NULL	
perna_esquerda	float	Yes	NULL	
ombro	float	Yes	NULL	
pescoco	float	Yes	NULL	
punho	float	Yes	NULL	
joelho	float	Yes	NULL	
tornozelo	float	Yes	NULL	
usuario_id	int(11)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	0	A	No	
fk_avaliacao_usuario1_idx	BTREE	No	No	usuario_id	0	A	No	

exercicio

Column	Type	Null	Default	Comments
id (Primary)	int(11)	No		
nome	varchar(128)	No		
grupo_muscular	varchar(256)	No		
ficha_id	int(11)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	10	A	No	
fk_exercicio_ficha1_idx	BTREE	No	No	ficha_id	2	A	No	

exercicio_treino

Column	Type	Null	Default	Comments
id (Primary)	int(11)	No		

nome	varchar(128)	No		
grupo_muscular	varchar(256)	No		
treino_id	int(11)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	10	A	No	
fk_exercicio_treino_treino1_idx	BTREE	No	No	treino_id	2	A	No	

ficha

Column	Type	Null	Default	Comments
id (Primary)	int(11)	No		
nome	varchar(64)	No		
dias_semana	varchar(256)	No		
usuario_id	int(11)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	2	A	No	
fk_ficha_usuario1_idx	BTREE	No	No	usuario_id	1	A	No	

serie

Column	Type	Null	Default	Comments
id (Primary)	int(11)	No		
tipo	enum('peso', 'tempo')	No		
repeticoes	int(11)	Yes	NULL	
peso	int(11)	Yes	NULL	
tempo	int(11)	Yes	NULL	
exercico_id	int(11)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	30	A	No	
fk_serie_exercicio_idx	BTREE	No	No	exercico_id	10	A	No	

serie_treino

Column	Type	Null	Default	Comments
id (Primary)	int(11)	No		
tipo	enum('peso', 'tempo')	No		
repeticoes	int(11)	Yes	NULL	
peso	int(11)	Yes	NULL	
tempo	int(11)	Yes	NULL	
feito	tinyint(1)	No		
exercico_treino_id	int(11)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	30	A	No	
fk_serie_treino_exercicio_treino1_idx	BTREE	No	No	exercico_treino_id	10	A	No	

treino

Column	Type	Null	Default	Comments
--------	------	------	---------	----------

id (<i>Primary</i>)	int(11)	No		
data	date	No		
hora_inicio	time	No		
hora_fim	time	Yes	NULL	
usuario_id	int(11)	No		
ficha_id	int(11)	Yes	NULL	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	2	A	No	
fk_treino_usuario1_idx	BTREE	No	No	usuario_id	1	A	No	

usuario

Column	Type	Null	Default	Comments
id (<i>Primary</i>)	int(11)	No		
nome	varchar(64)	No		
email	varchar(128)	No		
senha	varchar(128)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	2	A	No	
email	BTREE	Yes	No	email	2	A	No	