

## **PROPOSTA DE DESENVOLVIMENTO DE APLICATIVO MOBILE VOLTADO PARA TREINOS EM CASA**

Manoel Nazaro Da Silva Júnior  
Hercílio de Medeiros Sousa

### **1 INTRODUÇÃO**

A pandemia de covid 19 pegou a população mundial de surpresa, fazendo com que as pessoas mudassem de rotina completamene. Diante desse cenário, a atividade física, que é importante para a saúde e bem-estar do indivíduo, acabou sendo negligenciada e, em muitos casos, ficando fora da rotina das pessoas.

Tendo em vista oferecer auxílio para que pessoas adicionem exercícios ao seu dia-a-dia, pensou-se na elaboração de uma aplicação mobile voltada para treinamentos em casa. Diante disso, os usuários poderão se exercitar em casa sem a necessidade de se arriscar em academias ou praticar exercícios fora do ambiente familiar.

A prática de atividades físicas é aliada no combate de várias doenças, fortalece o sistema imunológico, ajuda no emagrecimento, diminui riscos de doença cardiovasculares, além de aumentar a circulação sanguínea cerebral e a produção de catecolaminas com a noradrelina e dopamina, que são as substâncias essenciais para a memória.

Mesmo existindo há décadas o esporte e atividade física continua no século XIX transportando as transformações políticas e sociais que começaram em séculos anteriores, comprovando, desde então, uma tendência a servir como projeção de dinâmica social. (SILVA *et al.*, 2007).

### **2 Fundamentação Teórica**

O cenário vivenciado nos dias atuais, de pandemia, por assim dizer, fortificou a ideia de que o ser humano é capaz de ser criativo, tomar decisões, trabalhar, concluir atividades e obter resultados de forma remota, utilizando aplicativos e ferramentas digitais que auxiliam na resolução de suas demandas.

A principal característica dos aplicativos móveis é a quebra da limitação da mobilidade, uma vez que os smartphones são como um computador de bolso, que

pode acompanhar o usuário 24 horas por dia. Outro aspecto relevante é a personalidade que o equipamento proporciona aos seus usuários, considerando que o profissional pode utilizar seu aparelho pessoal, com o qual já está acostumado a lidar diariamente. (TIBES; DIAS; ZEM-MASCARENHAS, 2014).

Sendo assim, até mesmo os exercícios físicos passam a exigir uma demanda tecnológica, dispensando os aparelhos encontrados em academias e utilizando o peso do corpo a partir de treinos guiados por aplicativos instalados em aparelhos móveis. De acordo com Kotarsky (2018), O treinamento com o peso corporal (também conhecido popularmente como calistenia) é um método de treinamento clássico e sua prática foi amplamente difundida desde o século XIX, mas pouco evidenciada na literatura científica ao longo 15 dos anos.

Esse tipo de exercício físico visa promover adaptações multissistêmica utilizando o peso do corpo como sobrecarga com nenhum ou poucos implementos”. Este atributo faz com que o exercício com o peso corporal se torne de fácil aplicabilidade, além de possuir uma ótima relação custo benefício quando comparado a outras possibilidades de treino que necessitam de máquinas ou materiais (THOMPSON, 2018).

## **2.1 Saúde e bem-estar**

A qualidade de vida e bem-estar implica mais que a ausência de déficits e maus tratos, ou seja, a qualidade de vida é um levantamento relevante e utilizado da medicina na saúde pública. (GASPAR *et al.*, 2006)

Não se deseja focalizar o subjetivismo que tanto a locução perfeição, como bem-estar trazem em seu bolso. Mas, ainda que as empregue a conceitos externos de avaliação, a perfeição não é notória. Se se trabalhar com um referencial objetivista, isto é, com uma avaliação do grau de perfeição, bem-estar ou felicidade de um sujeito é externa a ele próprio. (SEGRE; FERRAZ, 1997)

No século atual, tem-se notado uma transformação precoce no padrão de vida da sociedade. A mecanização, os avanços tecnológicos são presentes cada vez mais no cotidiano das pessoas, ou seja, esses mecanismos fazem com que as pessoas poupem esforços físicos, e conseqüentemente, ocorre a diminuição de atividade física. (NAHAS; BARROS; FRANCALACCI, 2000).

## 2.2 atividade física

O reconhecimento dos benefícios da prática de atividade física regular na melhoria da qualidade de vida chega a despertar uma enorme atenção quanto a complexa relação entre os níveis de prática das atividades físicas e a saúde das pessoas. (GUEDES D.; GUEDES J., 1995)

A atividade física é definida como qualquer movimento do corpo, elaborado pelos músculos esqueléticos, que resulta nos gastos energéticos e níveis de repouso. Assim, toda energia necessária para realização de movimentos corporais devem se manter ao nível de atividades físicas exigidas pelo movimento. (GUEDES D.; GUEDES J., 1995)

Os indícios destacam a repercussão positiva da atividade física regular em aspectos cognitivos, na saúde mental e no bem-estar dos indivíduos durante o envelhecimento, alguns chegam a destacar o efeito das atividades físicas, mais especificamente da caminhada, na diminuição de riscos cardiovasculares. (MATSUDO, 2009).

## 2.3 calistenia

A calistenia é um treino funcional com o objetivo de ganhar massa magra, resistência muscular e maior denotação no corpo. O treino possui como principais atividades a barra fixa, paralela e flexões e braço em determinadas variações. A calistenia tem como propósito trazer benefícios como: o aumento da hipertrofia muscular, melhor resistência muscular e cardiorrespiratória, aumento da flexibilidade e o equilíbrio corporal. Para a prática da calistenia não precisa de um local específico ou equipamentos como mostra na figura 1.

**Imagem 1** – Demonstração treinamento calistenia



**Fonte:** Iridium (2021).

A organização de uma sessão de calistenia obedece a 3 princípios básicos, sendo eles: seleção, precisão e totalidade consequentemente os exercícios são distribuídos em três grupos tais como introdutório, exercícios de aquecimentos composto por exercícios de extensão, flexão entre outros. (MACHADO, 2018).

A calistenia, na última década, tem ganhado espaço e atenção no campo mundo da educação física, tornando-se verdadeira febre “no mundo fitness”, onde de acordo com um levantamento feito pela Fitnes Trends em 2019 tal modalidade ocupa o top 10 de atividades praticadas no recorrente ano. (FARIAS, 2019).

## **2.4 java**

*Java* é uma linguagem de programação e um ecossistema computacional que foi criado pela sun Microsystems em meados dos anos 90. Posteriormente, foi comprado pela Oracle. Devido a probabilidade de escrever o código apenas uma única vez e conseguir rodar o sistema em vários dispositivos, a tecnologia tornou-se muito popular, passando a ser usada no mundo todo e praticamente em todos aparelhos eletrônicos. (MELO, 2021).

A linguagem Java chamou uma grande atenção da indústria e do meio acadêmico com a possibilidade de sua utilização para cálculos científicos. Os aplicativos nessa área estariam classificados como computação numéricas e geralmente tem requisitos altos para que possa desempenhar. (MOREIRA *et al.*, 2000).

De acordo com Indrusiak, (1996) a linguagem Java obtém êxito ao cumprir requisitos de sua diferenciação. Com o crescimento da internet os trabalhadores da sun perceberam que seria uma abertura para aplicar no comércio a linguagem de programação.

A partir dessa ideia, adaptaram o código Java para que fosse usado em sistemas de microcomputadores conectados a internet. Sendo assim, o Java proporcionou a criação de programas que tinha tráfegos de dados utilizando uma interface gráfica na web.

## **2.5 spring boot**

O *Spring boot* segundo Boaglio (2017) impulsionou o desenvolvimento de micro serviços, da mesma forma ajuda na configuração automática de todas as

dependências, desde 2003 o Spring teve um crescimento devastador na visão do desenvolvedor, assim, aumentando as opções e modelos de implementar códigos.

O projeto foi desenvolvido com o objetivo de criar uma aplicação de maneira simples e coloca-la no ar rapidamente. É um sistema que funciona sem depender de um servidor de aplicação, Spring é uma nova concepção, na qual acelera o desenvolvimento e simplifica bastante a vida do desenvolvedor. (BOAGLIO, 2017)

O *Spring boot* é a principal tecnologia usada em *Java* com intuito de gerar implementações em micro serviços. O Spring diminui consideravelmente o desenvolvimento do micro serviços quando minimiza as tarefas centrais da construção Rest que é baseada em micro serviços. (BATISTA; BASTOS, 2017)

## **2.6 banco de dados mysql-server**

O MySQL é um programa gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas preparado regular suas características de dados. O acesso opera a sistema SQL (Linguagem de Consulta Estruturada), quanto é a sistema mais popular para acrescentar, acessar e administrar o conteúdo armazenado num banco de dados. (PISA, 2012).

## **3 METODOLOGIA**

O presente trabalho tem como proposta o desenvolvimento de um sistema intitulado *Academy Calisthenics*, objetivando a introdução de tecnologias abordando procedimentos que serão utilizados no desenvolvimento da aplicação requisitos funcionais e não funcionais.

Para elaboração, possível aplicação e uso, levou-se em consideração os principais fatos ocorridos durante o ano de 2020-2021 ocasionados pela pandemia COVID19. Uma vez que a utilização do software pode contribuir para melhorar a saúde física, mental e estilo de vida da população que faça uso do mesmo.

### **3.1 Requisitos do sistema**

Quando fala-se de aplicação pensa-se logo o que esse software faz ou deveria fazer e aí onde entra os requisitos do sistema. Assim, o requisito do sistema consiste em descrever todos as atividades e serviços que o sistema irá realizar, entre eles nós temos requisitos funcionais e não funcionais.

### 3.2 Requisito funcionais

São comportamentos e ações ou serviços que o usuário irá realizar dentro do sistema, ou seja, e a forma como o sistema deve reagir para satisfazer o cliente. Ex: cadastrar um cliente.

**Tabela 1** - Requisitos Funcionais

| IDENTIFICADOR | NOME                                 | DESCRIÇÃO  | PRIORIDADE |
|---------------|--------------------------------------|--|------------|
| RF01          | Cadastrar Aluno                      | Sistema deve permitir cadastrar o aluno inserindo os dados; nome,cpf,email,whatsapp, horário de treino, tipo de treino, caso o individuo quiser cadastrar duas pessoas com o mesmo CPF o sistema irá barrar e trazer uma mensagem de erro. | ESSENCIAL  |
| RF02          | Consultar Aluno                      | Sistema deve realizar consulta pelo cliente usando o filtro CPF, caso CPF seja invalido trazer uma mensagem de erro.   | ESSENCIAL  |
| RF03          | Editar Aluno                         | Sistema irá permitir que o dado dos alunos seja alterado será pego pelo ID e será alterado, caso o ID seja invalido trazer uma mensagem de erro.   | DESEJAVEL  |
| RF04          | Listar Alunos                        | Sistema deve mostrar lista de aluno cadastrado em uma tabela, caso a lista esteja vazia trazer uma mensagem.   | ESSENCIAL  |
| RF05          | Excluir Aluno                        | Sistema permitirá que o aluno será excluído de forma logica, ele só será excluído do banco caso ele não queira mais utilizar a aplicação.  | ESSENCIAL  |
| RF06          | Emitir alerta de atraso de pagamento | Sistema deve emitir mensagem de alerta para o email do aluno caso ele passe do dia do vencimento.  | ESSENCIAL  |

|      |                     |   |            |
|------|---------------------|---|------------|
| RF07 | Cadastrar Professor | Sistema irá permitir cadastrar o professor com os seguintes dados; nome, CPF, hora, salário, e irá fazer uma verificação se o CPF já existe no banco de dados, caso exista o mesmo não poderá ser cadastrado, e trará uma mensagem de erro. | IMORTANTE  |
| RF08 | Editar Professor    | Sistema irá ditar os dados do aluno caso o aluno exista do banco, caso contrário trazer uma mensagem de erro.   | DESEJAVEL  |
| RF09 | Excluir Professor   | Sistema irá excluir o professor.  | ESSENCIAL  |
| RF10 | Listar Professores  | Será listado todos os professores cadastrados em ma tabela, caso não tenha nenhum professor cadastrado trazer uma mensagem.   | IMPORTANTE |

**Fonte:** Próprio Autor (2021).

Requisitos funcionais têm como principal objetivo mostra quais os comportamentos que o sistema terá e os papéis que ele desempenhará para satisfazer o usuário.

### 3.3 Requisitos não funcionais

Requisitos não funcionais são aqueles que não são funcionalidades do sistema, mas que se relacionam à forma que o sistema necessita para executado. Ex: tipo de sistema operacional que a aplicação realizará os serviços.

São restrições que o sistema deve ter para poder funcionar. Sem essas restrições é impossível que o sistema rode ou execute alguma funcionalidade.

**Tabela 2** - Requisitos não funcionais

| IDENTIFICADOR | NOME | DESCRIÇÃO  | PRIORIDADE |
|---------------|------|--|------------|
| RF01          | S.O  | Aplicação multiplataforma desenvolvida para trabalhar em vários s.os Ex; Windows, Linux, iOS, android, | IMPORTANTE |

|      |                               |   |           |
|------|-------------------------------|---|-----------|
|      |                               | com arquitetura x86 e x64.  |           |
| RF02 | Hardware                      | Aplicação deve ter no mínimo 4G de RAM, placa-mãe gigabyte, processador a partir do i3 4ª geração.  | ESSENCIAL |
| RF03 | Desenvolvimento & Tecnologias | Aplicação será desenvolvido com Angular no front-end, e java com Spring no back-end, SGBD usado será o MYSQL.   | ESSENCIAL |
| RF04 | IDEs                          | Será utilizado como ambiente de desenvolvimento as IDEs ; Visual Studio Code como ambiente de desenvolvimento front-end, e o Eclipse como ambiente de desenvolvimento back-end. | ESSENCIAL |
| RF05 | SGBDs                         | Aplicação irá utilizar o MYSQL SERVER como base de dados, para manipular informações.   | ESSENCIAL |
| RF06 | Manutenção                    | Sistemas ira permitir atualização de 6 em 6 meses e reparos.  | DESEJAVEL |

**Fonte:** Próprio Autor (2021).

Requisitos não funcionais tem o objetivo de mostrar o que o sistema precisa para poder funcionar e executar as funcionalidades.

## 4 DESENVOLVIMENTO

O presente trabalho se propõe a construir um sistema intitulado *Academy Calisthenics*. O sistema terá funcionalidades de cadastrar todos os dados dos alunos, em um banco de dados e assim gerar um perfil do aluno, assim como o tipo de aula que o mesmo irá ter de acordo com o seu objetivo.

De acordo com a figura 1 temos a classe modelo que com seus atributos poderá ser persistido e transformara as propriedades em colunas no banco e a entidade aluno será tabela, podemos ver que o id será gerado de forma automática.

**Figura 1** - classe modelo

```

14
15 @Table(name = "alunos")
16 @Entity
17 @Getter
18 @Setter
19 public class Aluno implements Serializable{
20
21
22     private static final long serialVersionUID = 1L;
23
24     @Id
25     @GeneratedValue(strategy = GenerationType.AUTO)
26     private Long id;
27
28     @Column
29     private String nome;
30
31     @Column
32     private String cpf;
33
34     @Column
35     private String hora;
36
37     @Column
38     private String whatsapp;
39

```

**Fonte:** Próprio autor (2021).

Conforme podemos observar, a figura 2 contém a classe que representa um repositório de uma interface que é estendida pelo JPA que irá conter todos os métodos em que compõe a funcionalidade do sistema como vemos na figura 2 abaixo.

**Figura 2** - classes modelo

```

J AlunoRepository.java
1 package tcc.com.br.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5
6
7
8 @Repository
9 public interface AlunoRepository extends JpaRepository<Aluno, Long>{
10
11
12     public boolean existsByCpf(String cpf);
13 }
14

```

**Fonte:** Próprio Autor (2021).

Na classe controladora irá acontecer toda comunicação HTTP e na qual o cliente irá enviar uma requisição a classe irá pegar essa requisição e retornar uma resposta em HTTP, como vemos na figura 3, abaixo.

**Figura 3 - classe controller aluno**

```
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import tcc.com.br.model.Aluno;
import tcc.com.br.service.AlunoService;

@RestController
@RequestMapping("/alunos")
public class AlunoController {

    @Autowired
    private AlunoService alunoService;

    @PostMapping("/salvar")
    public ResponseEntity<Aluno> salvar(@RequestBody Aluno aluno){
        return new ResponseEntity<>(alunoService.salvarAluno(aluno), HttpStatus.CREATED);
    }
}
```

**Fonte:** Próprio Autor (2021).

A Classe de serviço é a classe onde fica toda a regra de negócio executada pelo sistema será implementado aqui como vemos na figura 4 na hora de adicionar um aluno ao repositório será feita uma verificação antes do CPF do aluno se esse CPF existir dentro do banco de dados ele não será salvo, por outro lado se o CPF não existir no banco o mesmo será adicionado.

**Figura 4 - classe controller serviços**

```
@Service
public class AlunoService {

    @Autowired
    private AlunoRepository alunoRepository;

    public Aluno salvarAluno(Aluno aluno) {
        if(!alunoRepository.existsByCpf(aluno.getCpf())) {
            return alunoRepository.save(aluno);
        }
        System.out.println("CPF já existe!");
        return null;
    }
}
```

**Fonte:** Próprio Autor (2021).

Tem-se a classe modelo aula e seus atributos como vimos e vemos também que a classe tem uma propriedade aluno se relacionando com a entidade aula, ou seja, temos um relacionamento de composição na figura 5.

**Figura 5** - classe modelo

```

@Getter
@Setter
public class Aula implements Serializable{

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @ManyToOne
    @JoinColumn(name = "aluno_id")
    private Aluno aluno;

    @Column
    private String duracao;

    @Column
    private String tipoTreino;
}

```

**Fonte:** Próprio Autor (2021).

No serviço será criado a aula e a descrição de como aconteceu tudo se é quem será o participante de forma detalhada e depois será salvo no banco de dados tudo como podemos observar na imagem 6.

**Figura 6** - classe de serviço

```

@Autowired
private AulaRepository aulaRepository;

// @Autowired
// private AlunoRepository alunoRepository;

public Aula salvarAula(Aula aula) {
    Aluno aluno = new Aluno();
    if(aula.getAluno() != null) {
        String descricao = "Aula agendada para " + aluno.getHora() +
            aula.getTipoTreino() + " com a duração de " + aula.getDuracao();
        return aulaRepository.save(aula);
    }
    return null;
}
}

```

**Fonte:** Próprio Autor (2021).

Por sua vez o controller vai processar as informações vinda do front-end , e irá tratar a requisição e retornará uma resposta HTTP para o usuário como mostra a figura 7.

**Figura 7 - classe controller**

```
import org.springframework.beans.factory.annotation.Autowired;

@RestController
@RequestMapping("/aula")
public class AulaController {

    @Autowired
    private AulaService aulaService;

    @PostMapping("/salvar-aula")
    public ResponseEntity<Aula> gerarAula(@RequestBody Aula aula){
        return new ResponseEntity<>(aulaService.salvarAula(aula),HttpStatus.CREATED);
    }
}
```

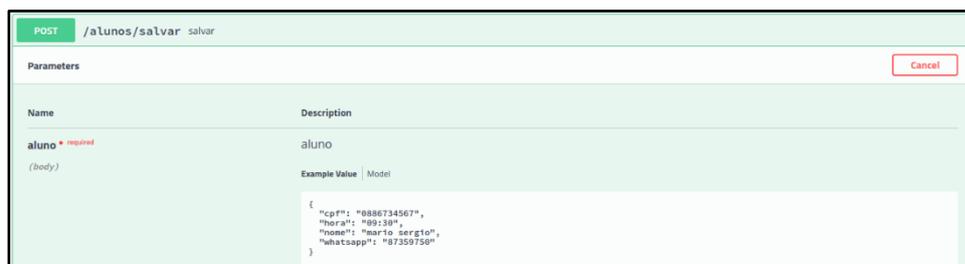
**Fonte:** Próprio Autor (2021).

O Swagger é uma ferramenta utilizada para documentar APIs criando interações com uma interface amigável ao usuário simulando um front-end como mostra a figura 8.

**Figura 8 - Swagger**

**Fonte:** Próprio Autor (2021).

Utilizando o Swagger para cadastra o aluno temos e só colocar as informações na parte inferior do corpo do Swagger veja a seguir na figura 9.

**Figura 9 - Swagger (cadastrando aluno)**

**Fonte:** Próprio Autor (2021).

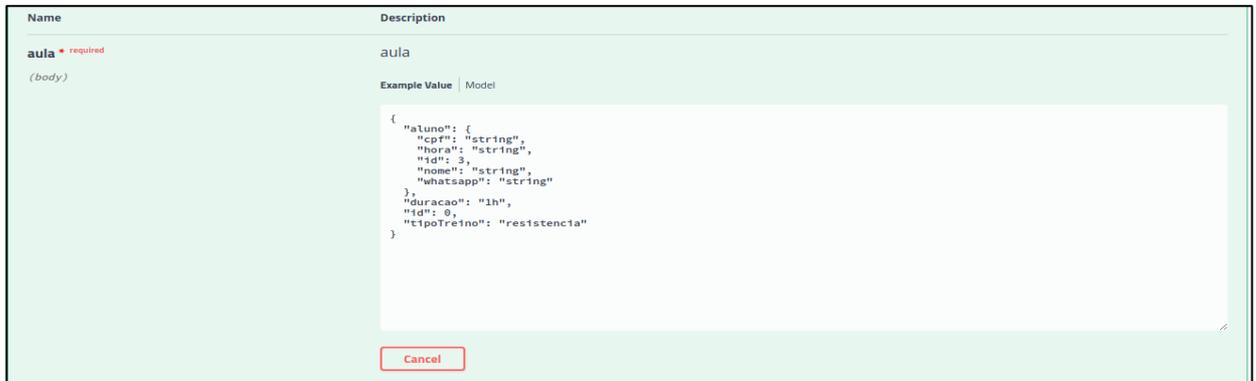
Como pode-se observar na figura 10 os Dados estão sendo enviados para o banco de dados.

**Figura 10 - Swagger JSON detalhes**

**Fonte:** Próprio Autor (2021).

E passado no Json no os valores tipo Treino no atributo aluno e passado o id do aluno e matriz de aluno será preenchida de forma automática como podemos observar nas figuras 11 e 12.

**Figura 11 - Swagger JSON**



Fonte: Próprio Autor (2021).

**Figura 12 - Swagger JSON detalhes**



Fonte: Próprio Autor (2021).

## 9 CONSIDERAÇÕES FINAIS

O estudo teve como objetivo o desenvolvimento e possível aplicação de um software que possa suprir a necessidade de pessoas que gostam de se exercitar e praticar atividades físicas. Com os resultados do estudo pode-se dizer que o uso de tal software pode ajudar a comunidade que estejam em isolamento social possam manter uma rotina de treinos, evitando espaços com maior número de pessoas, o que pode gerar insegura devido as aglomerações.

## REFERÊNCIAS

BOAGLIO, Fernando. **Spring Boot: Acelere o desenvolvimento de microsserviços**. São Paulo: Casa do Código, 2017.

BATISTA, Mateus Alfredo; BASTOS, William Carvalho. **Desenvolvendo aplicações web baseadas em microserviço utilizando o Framework Spring**. 2017. 124 f. Monografia (Graduação Sistemas de Informação) - Universidade do Sul de Santa Catarina Mateus Alfredo Batista William Carvalho Bastos, Palhoça, 2017.

FARIAS, Thalles. **Desvendando a origem da calistenia**. 2019. Disponível em: <https://teammaf.com.br/wpcontent/f15d/uploads/teammaf.com.br/2019/08/-ORIGEM-DA-CALISTENIA-1.pdf>. Acesso em: 20 maio 2021.

GASPAR, Tania et al. Qualidade de vida e bem-estar em crianças e adolescentes. **Revista Brasileira de Terapias Cognitivas**, v. 2, n. 2, p. 47-60, 2006.

GUEDES, Dartagnan Pinto; GUEDES, Joana Elisabete Ribeiro Pinto. Atividade física, aptidão física e saúde. **Revista Brasileira de Atividade Física & Saúde**, v. 1, n. 1, p. 18-35, 1995.

INDRUSIAK, Leandro Soares. **Linguagem Java**. 1996. Disponível em: <https://www.cin.ufpe.br/~arfs/introjava.pdf>. Acesso em: 22 maio 2021.

IRIDIUM. **A calistenia pode ser muito útil para os marombas em busca de um shape insano. E a gente vai mostrar aqui algumas razões para você acreditar nisso**. 2020. Disponível em: <https://www.feitodeiridium.com.br/razoes-colocar-calistenia-treino/>. Acesso em: 22 abr. 2021.

KOTARSKY, Christopher J. et al. Effect of progressive calisthenic push-up training on muscle strength and thickness. **The Journal of Strength & Conditioning Research**, v. 32, n. 3, p. 651-659, 2018.

MATSUDO, Sandra Marcela Mahecha. Envelhecimento, atividade física e saúde. **BIS. Boletim do Instituto de Saúde (Impresso)**, n. 47, p. 76-79, 2009.

MACHADO, Alexandre Fernandes. **A nova calistenia**. 2018. Disponível em: <https://www.portalatlanticaeditora.com.br/index.php/revistafisiologia/article/view/2463> Acesso em: 20 maio 2021.

MELO, Diego. **O que é Java?** 2021. Disponível em: <https://tecnoblog.net/416833/o-que-e-java-guia-para-iniciantes/amp/>. Acesso em: 22 maio 2021.

MOREIRA, Je *et al.* **Programação Java para computação numérica de alto desempenho**. 2000. Disponível em: <https://ieeexplore.ieee.org/abstract/document/5387068>. Acesso em: 22 maio 2021.

NAHAS, Markus Vinicius; BARROS, Mauro V.G; FRANCALACCI, Vanessa. O pentágulo do bem-estar-base conceitual para avaliação do estilo de vida de indivíduos ou grupos. **Revista Brasileira de Atividade Física & Saúde**, v. 5, n. 2, p. 48-59, 2000.

PISA, Pedro. **O que é e como usar o MySQL?** 2020. Disponível em: <https://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>. Acesso em: 31 maio 2021.

SILVA, Rodrigo Sinnott *et al.* Atividade física e qualidade de vida. **Ciência & Saúde Coletiva**, v. 15, p. 115-120, 2010.

SEGRE, Marco; FERRAZ, Flávio Carvalho. **O conceito de saúde**. 1997. Disponível em: <https://www.scielo.org/article/rsp/1997.v31n5/538-542/pt/>. Acesso em: 20 maio 2021.

TIBES, Chris Mayara dos Santos; DIAS, Jessica David; ZEM-MASCARENHAS, Silvia Helena. Aplicativos móveis desenvolvidos para a área da saúde no Brasil: revisão integrativa da literatura. **Revista Mineira de Enfermagem**, v. 18, n. 2, p. 471-486, 2014.

THOMPSON, W. R. Worldwide survey of fitness trends for 2018. **ACSM's Health & Fitness Journal**. v. 21. n. 6. p. 10-19. 2017.