**CENTRO PAULA SOUZA**

**ETEC PEDRO FERREIRA ALVES**

**Desenvolvimento de Sistemas Integrado ao Ensino Médio**

**Deivid Maicon da Silva**

**Felipe Rebeck Pocobello Santos**

**Gabriel Matheus Crispim de Souza**

**Gustavo Dionysio Tamashiro**

**Maria Eduarda Silva Oliveira**

**APLICATIVO ACADEMIA ESTHETIC-FIT**

**Mogi Mirim**

**2022**

**Deivid Maicon da Silva**

**Felipe Rebeck Pocobello Santos**

**Gabriel Matheus Crispim De Souza**

**Gustavo Dionysio Tamashiro**

**Maria Eduarda Silva Oliveira**

**APLICATIVO ACADEMIA ESTHETIC-FIT**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas Integrado ao Ensino Médio da Etec Pedro Ferreira Alves orientado pelo Prof. Marcos Roberto de Moraes e Prof. Rafael da Silva Polato.

**Mogi Mirim**

**2022**

*“O Estado preocupa-se com a saúde do indivíduo em função de sua utilização como instrumento de trabalho e não em função de suas esperanças, de seus anseios, de seus temores ou de seus sofrimentos.”*

JAYME LANDMANN

**RESUMO**

Seguindo o pensamento de Jean Fourastié, que afirmou: “Nada será menos industrial que a civilização nascida da Revolução Industrial”, a chegada da Revolução Industrial inicializada no século XVIII trouxe consigo a evolução do conhecimento acerca da sociedade, melhorando e visando o maior alcance do bom negócio, incluindo a Revolução Tecnológica que trouxe evoluções significativas para o convívio social, além de auxiliar substancialmente os avanços em outras áreas, como a saúde. Diante disso, com a chegada do ambiente pandêmico a tecnologia se tornou componente essencial para o convívio humano, trazendo o conceito de presença mesmo remotamente, porém com a chegada dele muitos sofreram com a escassez da tecnologia, principalmente ambientes não otimizados. Diante disso, a criação do software responsivo para Academia “Boa Forma” localizada em Santo Antônio de Posse, São Paulo, possui como propósito a melhoria de comunicação e automação do sistema de monitoramento e cadastro atual do local. Em vista do enfrentamento pandêmico nos anos de 2019 a 2021, a falta de otimização do sistema de cadastro se propôs disfuncional para o funcionamento do comércio, dessa forma, a solicitação de um software criado para atender os requisitos solicitados por tais, atendendo como cadastro, manipulação de dados, avisos sobre o local, informações de contato juntamente com ficha técnica dos usuários habituais do estabelecimento se propôs como solução do problema enfrentado, garantindo a proteção de qualquer tipo de contaminação por métodos antigos utilizados atualmente pelo sistema, optando de um sistema manual para um sistema virtual com o auxílio dos profissionais de TI do Curso de Desenvolvimento de Sistemas da ETEC Pedro Ferreira Alves, localizada em Mogi Mirim, São Paulo.

Palavras-chave: Pandemia, Tecnologia, Software, Comunicação, Negócios, Revolução Tecnológica, Convívio, Saúde.

**ABSTRACT**

Following the thought of Jean Fourastié, who stated: "Nothing will be less industrial than the civilization born of the Industrial Revolution", the arrival of the Industrial Revolution started in the 18th century brought with it the evolution of knowledge about society, improving and aiming at the greater reach of good business, including the Technological Revolution that brought significant developments for social coexistence, besides substantially assisting the advances in other areas, such as health. Given this, with the arrival of the pandemic environment, technology became an essential component for human coexistence, bringing the concept of presence even remotely, but with its arrival many suffered from the scarcity of technology, especially non-optimized environments. Given this, the creation of responsive software for Academia "Boa Forma" located in Santo Antônio de Posse, São Paulo, has as purpose the improvement of communication and automation of the monitoring system and current registration of the place. In view of the pandemic facing in the years 2019 to 2021, the lack of optimization of the registration system was proposed dysfunctional for the operation of the trade, thus, the request for a software created to meet the requirements requested by such, serving as registration, data manipulation, notices about the place, contact information along with the technical sheet of the usual users of the establishment was proposed as a solution to the problem faced, ensuring the protection of any kind of contamination by old methods currently used by the system, opting for a manual system for a virtual system with the help of IT professionals from the Systems Development Course of ETEC Pedro Ferreira Alves, located in Mogi Mirim, São Paulo.

Keywords: Pandemic, Technology, Software, Communication, Business, Technological Revolution, Socializing, Health.

**LISTA DE FIGURAS**

[Figura 1 - Exemplo de Código em PHP 12](#_Toc118792038)

[Figura 2 - Exemplo de Código JavaScript 13](#_Toc118792039)

[Figura 3 - Exemplo de Código Java 14](#_Toc118792040)

[Figura 4 - Exemplo de Código Ruby 15](#_Toc118792041)

[Figura 5 - Exemplo de Código Python 16](#_Toc118792042)

[Figura 6 - Diagrama de Caso de Uso UML 29](#_Toc118792043)

[Figura 7 - Home Screen 30](#_Toc118792044)

[Figura 8 - Protótipo Login Screen 31](#_Toc118792045)

[Figura 9 - Protótipo New Account Screen 31](#_Toc118792046)

[Figura 10 - Protótipo Planes Screen 32](#_Toc118792047)

[Figura 11 - Protótipo Info Screen 33](#_Toc118792048)

[Figura 12 - Protótipo Contact Screen 34](#_Toc118792049)

SUMÁRIO

[1 INTRODUÇÃO 8](#_Toc118790063)

[2 REVISÃO BIBLIOGRÁFICA 10](#_Toc118790064)

[2.1 Linguagens de Programação 10](#_Toc118790065)

[**2.1.1 PHP 10**](#_Toc118790066)

[**2.1.2 JavaScript 11**](#_Toc118790067)

[**2.1.3 Java 12**](#_Toc118790068)

[**2.1.4 Ruby 13**](#_Toc118790069)

[**2.1.5 Python 14**](#_Toc118790070)

[**2.1.6 Swift 15**](#_Toc118790071)

[2.2 Plataformas de Desenvolvimento 16](#_Toc118790072)

[**2.2.1 Web 16**](#_Toc118790073)

[**2.2.1.1 Front-end 16**](#_Toc118790074)

[**2.2.1.2 HTML 16**](#_Toc118790075)

[**2.2.1.3 CSS 17**](#_Toc118790076)

[**2.2.1.4 Conjunto de HTML, CSS e JavaScript 17**](#_Toc118790077)

[**2.2.2 Back-end 18**](#_Toc118790078)

[**2.2.3 Desktop 18**](#_Toc118790079)

[**2.2.3.1 React Native 19**](#_Toc118790080)

[**2.2.3.2 Xamarin 19**](#_Toc118790081)

[**2.2.3.3 Swiftic 20**](#_Toc118790082)

[**2.2.3.4 Ruby on Rails 20**](#_Toc118790083)

[**2.2.3.5 Express.js 20**](#_Toc118790084)

[**2.2.3.6 Laravel 21**](#_Toc118790085)

[**2.2.4 Plataformas Mobile 22**](#_Toc118790086)

[**2.2.4.1 Android 22**](#_Toc118790087)

[**2.2.4.2 iOS 22**](#_Toc118790088)

[**2.2.4.3 Harmony OS 23**](#_Toc118790089)

[2.3 Paradigmas de Programação 24](#_Toc118790090)

[**2.3.1 Paradigma Imperativo 24**](#_Toc118790091)

[**2.3.2 Paradigma Declarativo 24**](#_Toc118790092)

[**2.3.3 Paradigma Funcional 25**](#_Toc118790093)

[**2.3.4 Paradigma Lógico 25**](#_Toc118790094)

[**2.3.5 Paradigma Orientado a Objetos 25**](#_Toc118790095)

[**2.3.6 Paradigma Orientado a Eventos 26**](#_Toc118790096)

[2.4 Engenharia de Software 26](#_Toc118790097)

[**2.4.1 UML 26**](#_Toc118790098)

[**2.4.2 Diagrama de Classes 27**](#_Toc118790099)

[**2.4.3 Diagramas Estruturais 27**](#_Toc118790100)

[**2.4.4 Diagramas Comportamentais 27**](#_Toc118790101)

[3 DESENVOLVIMENTO 29](#_Toc118790102)

[4 CONCLUSÃO 35](#_Toc118790103)

[REFERÊNCIAS 36](#_Toc118790104)

# 

# 1 INTRODUÇÃO

Desde a revolução tecnológica iniciada no século XX, a internet modificou a perspectiva do mundo, alterando o andamento de tecnologias que avançavam a demanda de trabalhos por aparelhos tecnológicos para os negócios. A demanda do mercado por tecnologias de negócios se tornou nítida e essencial, contudo, o desuso dessas mesmas ocasiona o retrocesso de negócios, resultando ao atraso de investimentos, aumentando os gastos por tais.

Ao mencionar a modernização dos métodos de trabalho antigos pela utilização de aparelhos tecnológicos evoluiu a fluidez dos lucros nos meios dos negócios, juntamente trabalhando o avanço de troca de informação em diferentes ambientes, onde o uso desses aparelhos para o acesso de informações diversas da vida cotidiana dos cidadãos se tornou mais ativo, como o acesso a aplicativos de saúde física.

A participação dos aparelhos tecnológicos em atividades físicas trouxe consigo a facilidade de troca de informações entre profissionais da saúde com seus clientes. Tendo em vista o cenário pandêmico enfrentado, o uso dos aparelhos tecnológicos acabou por ser extremamente necessário, mesmo após o retorno parcial das atividades, os trabalhos presenciais acabaram por serem substituídos por aplicativos e sites gerenciados pelos profissionais de software.

No cotidiano a relação entre a tecnologia e a saúde física têm sido vistas como antônima, tendo uma visão geral que com a evolução da tecnologia, o sedentarismo também evoluiu. Entretanto, com novos dados levantados, houve um novo consenso, onde a aplicação da tecnologia no ambiente de saúde trará a progresso para a população que possui dificuldade de manusear as duas áreas cotidianas juntas, dessa forma, com a criação de um aplicativo de treino com equipamentos de monitoramento auxiliarão para a melhoria da saúde física utilizando a tecnologia como principal agente de auxílio da população.

**1.1 Justificativa**

Tendo em vista o ambiente pandêmico, a problematização do contato físico por conta da transmissão do vírus se tornou extremamente agravada, de forma que muitos indivíduos ficassem indispostos com medo de obterem a doença por tal método de contaminação. Com a utilização de métodos antigos de fichas podendo ocorrer a transmissão, houve retrocesso do negócio aumentando os gastos por métodos menos atuais de comunicação entre o cliente e o profissional. Dessa forma com a criação de uma aplicação para auxiliar os clientes que retomam as atividades físicas presenciais contando com a facilidade da otimização do sistema, gerenciando seus próprios treinos por meio de seu dispositivo móvel. Além de evitar a contaminação durante a permanência no local de suas atividades físicas, contando com a enriquecimento de comunicação com o profissional da área de educação física, disponível no estabelecimento.

**1.2 Objetivo Geral**

Otimizar o sistema de cadastro e manutenção dos treinos estabelecidos para os alunos da academia Boa Forma, otimizando e facilitando o trabalho manual através de um sistema virtual.

**1.3 Objetivo Específico**

1. Demonstrar informações sobre a academia aos futuros alunos;
2. Cadastrar novos alunos e manusear os dados no sistema;
3. Verificar os objetivos dos alunos para elaboração de didáticas físicas;
4. Estabelecer plano e prazo diante do objetivo designado;
5. Elaborar didáticas físicas especificadas pelo professor;
6. Auxiliar na aplicação das didáticas físicas a cada aluno;
7. Acompanhar o desenvolvimento do condicionamento físico ao longo do prazo estabelecido;

**1.4 Hipóteses**

1. A criação de um sistema otimizado de cadastro e manutenção de dados os alunos ficaram mais propostos a melhorarem sua saúde física.
2. A criação de fichas de treino virtuais é mais fácil de manusear os treinos dos diferentes alunos frequentes da academia local.
3. O acompanhamento diário com os professores da academia traz a melhoria da comunicação entre os alunos e os professores.
4. A manutenção virtual das fichas de treino facilita o trabalho de atualização de dados dos professores.

# REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentado informações sobre os componentes utilizados e necessários para o desenvolvimento da aplicação, apresentando as linguagens de Programação, Plataformas de Desenvolvimento, Paradigmas de Programação e Engenharia de Software com o intuito de justificar as ferramentas utilizadas para o desenvolvimento do Trabalho de Conclusão de Curso (TCC).

## Linguagens de Programação

Segundo Estrella (2022), as Linguagens de Programação ou Linguagens de *Scripts* são determinadas linguagens de *scripts* que possuem função de automatizar a execução de tarefas em um ambiente de tempo para execução especial, inclui dizer que para a construção de uma página estática, construída com HTML e CSS, é necessário realizar ações especificas determinadas de acordo com o *script*. As linguagens podem rodar em dois ambientes divergentes, sendo Back-End e Front-End. São exemplos de linguagens de programação que funcionam nos dois ambientes:

* PHP;
* JavaScript;
* Java;
* Ruby;
* Python;
* Swift.

### 2.1.1 PHP

Segundo Estrella (2022), PHP é a sigla denominada ao PHP: *Hypertext Preprocessor* (Pré-Processador de Hipertexto), originalmente chamado de *Personal Home Page* (Página Inicial Pessoal). Sendo uma linguagem de programação voltada para o desenvolvimento de aplicações para a web e para criação de sites, interligando os servidores com a interface do usuário.

Atualmente, o PHP está em sua versão 7.2, entretanto uma das versões mais utilizadas pelos programadores é sua versão 5.0, agregando a possibilidade de implementar *handlers* ou apontadores, um erro notado na sua versão 4, que constava obtendo impossibilidade de criar cópias de objeto, corrigida em sua versão 5.

Segundo o site DevMedia (2008), uma das maiores vantagens dessa linguagem é a habilidade de integração em um arquivo HTML, atribuindo em conjunto que o programador pode utilizar marcações em HTML repetidamente sem necessidade de reescrever novamente. Demonstrado na Figura 1 há um exemplo de código em PHP.

Figura 1 - Exemplo de Código em PHP

**

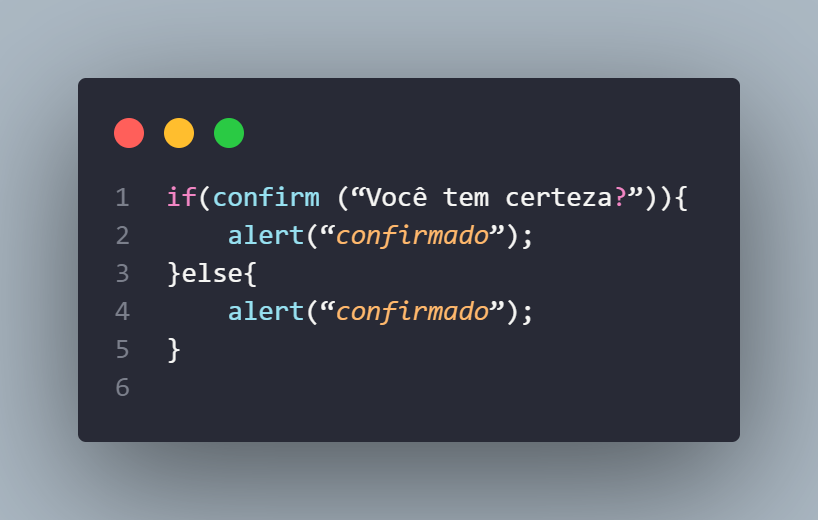
Fonte: Imagem dos Autores, 2022.

### 2.1.2 JavaScript

Segundo Roveda (2020) e Estrella (2021), JS é a sigla dada a JavaScript, sendo uma linguagem de programação de alto nível com finalidade de ser executada em navegadores e manipular comportamentos de páginas web, além de ter foco especializado na programação *client-size*, ou seja, sendo executada pelo navegador do usuário sem necessidade de transportação para outro ambiente.

De acordo com Melo (2021), ela é uma linguagem multiparadigma possuindo uma programação funcional e imperativa, versátil com tipagem dinâmica em conjunto de sintaxe acessível, além de possuir recursos avançados como orientação a objetos e APIs para acesso de textos, matrizes, datas e expressões regulares. O JS pode ser implementado em diferentes áreas de desenvolvimento, como Desenvolvimento Web, Desenvolvimento Mobile, Desktop e Jogos Eletrônicos. Na Figura 2 obtemos um exemplo de código em JavaScript:

Figura 2 - Exemplo de Código JavaScript



Fonte: Imagem dos Autores, 2022.

### 2.1.3 Java

Java é uma linguagem de programação orientada a objetos de extrema popularidade onde, a possibilidade de escrever o código apenas uma única vez e rodá-lo em diferentes dispositivos. O seu propósito é projetar códigos que podem controlar até o hardware, podendo citar não só computadores, mas diversos aparelhos e eletrodomésticos. (SITE MENTORAMA, 2021).

Segundo Melo (2021), a linguagem Java acaba sendo uma linguagem baseada em Classes e Orientação a Objetos possuindo foco em segurança, portabilidade e alta performance. Outras de suas características que se destacam é a similaridade de sintaxe a C/C++ com biblioteca de rotinas e APIs para trabalhar com recursos de rede juntamente com gerenciamento automático de memória.

Além de ser uma linguagem, o Java abriga a função de plataforma de software, sendo diferenciado em três tipos diferentes que abrigam as ferramentas necessárias para criação e execução de softwares e sistemas, sendo classificadas em alguns tipos:

* Java SE (*Java Platform, Standard Edition*): plataforma base possuindo bibliotecas comuns e o ambiente padrão de execução;
* Java EE (*Java Platform, Enterprise Edition*): possuindo recursos adicionais para desenvolvimento de aplicações corporativas;
* Java ME (*Java Platform, Micro Edition*): foco para aplicações móveis e de sistemas embarcados.

Demonstrado na Figura 3 um exemplo de código em Java.

Figura 3 - Exemplo de Código Java

**

Fonte: Imagem dos Autores, 2022.

### 2.1.4 Ruby

De acordo com BratFisch (2021), Ruby é uma linguagem de programação orientada para objetos, além de ser umas das linguagens de código aberto e multiplataforma, podendo ser utilizada em diversos sistemas operacionais como Windows, Linux, macOS, entre outros. Essa linguagem foi inspirada em outras como Perl, Eiffel, LISP e Python.

Uma das características da linguagem Ruby que se destaca é por ser uma linguagem de simples compreendimento, possuindo uma semântica simples, além de garantir um grande bônus, sendo o aumento de produtividade aos programadores que a utilizam.

De acordo com Thiago (2008), Ruby é uma linguagem ainda nova no mercado, criada no ano de 1995 por Yukihiro Matsumoto, este que se baseou em criar uma linguagem de *script* mais poderosa do que Perl e mais orientada a objetos do que Python assim o profissional criou a linguagem Ruby, possuindo como atualização mais recente em sua versão 3.0.0.

Utiliza-se principalmente o desenvolvimento de aplicações web, na área de Back-end, Ruby acaba sendo umas das linguagens mais utilizadas pelos profissionais de programação, além de ser trabalhado também no Desenvolvimento Desktop. Um dos pontos que fazem essa linguagem ganhar grande destaque é o *framework* utilizado para a manutenção da mesma, o famoso Ruby on Rails que trabalha em conjunto de bibliotecas que facilitam a criação de aplicações, além de ter o Ruby Gems, outra ferramenta utilizada para o gerenciamento de pacotes desenvolvido pela comunidade Ruby. (NOLETO, 2020) e (ANDRADE, 2020).

A seguir na Figura 4 é citado um exemplo de código em Ruby.

Figura 4 - Exemplo de Código Ruby

Tela de computador com imagem de jogo de vídeo game

Descrição gerada automaticamente

Fonte: Imagem do Autores, 2022.

### 2.1.5 Python

De acordo com Roveda (2020), Python é uma linguagem de programação de alto nível, denominada também como *High Level Language*, sendo uma das linguagens mais próximas da língua humana, esta que acaba sendo uma das mais utilizadas, principalmente no Desenvolvimento Web, tanto no Front-end como no Back-end.

Além de possuir outras características marcantes para os programadores a utilizarem tanto, por possuir características como Dinâmica, Interpretada, Modular, Multiplataforma e Orientada a Objetos.

Ainda de acordo com Roveda (2020), por possuir uma sintaxe simples e de fácil compreendimento, essa linguagem acaba sendo utilizada não apenas para os profissionais de Tecnologia, mas como também aos Engenheiros, matemáticos, cientistas de dados, pesquisadores, entre outros.

Citado por Melo (2021), Python contribui com uma sintaxe capaz de agregar em conjunto o uso de bibliotecas sendo nativas e de terceiros, sendo muito útil em questão de mesclagem de tecnologias para o desenvolvimento.

Suas versões são baseadas em versões principais, onde a mais recente é 3.9 onde a *Python Software Foundation* oferece suporte para versões mais antigas, pois há grande divergência de atualizações entre as versões 2.x a 3.x.

Na Figura 5 um exemplo de código em Python:

Figura 5 - Exemplo de Código Python



Fonte: Imagem dos Autores, 2022.

### 2.1.6 Swift

Swift é uma linguagem de programação voltada para iOS, macOS, watchOS, tvOS e Linux desenvolvida pela Apple, é uma das linguagens mais recentes do mercado de desenvolvimento. (BEDRAN, 2021).

Ainda de acordo com Bedran (2021), Swift é inspirada em algumas linguagens, como Objective-C, Rust, Haskell, Ruby, Python, C#, entre outras. Sua primeira versão foi lançada em 9 de setembro de 2014 com a versão Gold Master do Xcode 6.0 para iOS, atualmente está na versão 5.4 lançada 26 de abril de 2021.

Duas características que mais ganham destaque na Swift é por ser uma linguagem de sintaxe simples e de código aberto, além de possuir seu próprio ambiente de desenvolvimento criado também pela Apple, o Swift Playgrounds, possuindo uma interface que consiga testar os códigos sem que haja a intervenção do desenvolvimento completo do projeto. (BEDRAN, 2021).

Ainda acordo com Bedran (2021), é citado outras características que não possuíam na Objective-C que é encontrada na Swift, são:

* Controle de Acesso;
* Tipagem de Valor;
* Enumeração e Casamento de Padrões;
* Tipos Opcionais;
* Tratamento de Erros e Iteradores e Clausuras.

## Plataformas de Desenvolvimento

### 2.2.1 Web

Segundo UseMobile (2022) e Roveda (2020), o Desenvolvimento Web é a área da programação responsável para a construção de sites, aplicativos e softwares em conjunto de outras ferramentas responsáveis por construírem a visualização atual da internet. Dentre as plataformas de Desenvolvimento possuímos diferentes ambientes citados anteriormente, sendo detalhados com mais afinco.

### 2.2.1.1 Front-end

Segundo Roveda (2020), o Front-end é uma área da programação voltada para a performance visual do site, ou seja, a área responsável por interação com o usuário, especificamente dizemos que é sua interface externa.

O programador responsável por dominar o Front-end compreende algumas linguagens de programação para o desenvolvimento de sua interface por meio de códigos, as principais linguagens de Front-end são a trindade como HTML, CSS e JavaScript. Além de possuir consigo a utilização de bibliotecas entre os profissionais do campo, sendo React, JQuery e Node.js. (ROVEDA, 2020) e (SOUTO, 2019).

### 2.2.1.2 HTML

Segundo Marques (2019), o HTML é uma Linguagem de Marcação de Hipertexto, sendo reconhecido como um componente básico da programação Web, permitindo inserir o conteúdo e estabelecer a estrutura básica de um *Website*, permanecendo uma organização no conteúdo estabelecido pelo programador.

Linguagem de marcação é composta por hipertextos, esses que são conjuntos de elementos conectados, como por exemplo eles podem ser palavras, imagens, vídeos e documentos, além de outros adereços contendo algum tipo de informação. Quando conectados formam uma rede de informações que permitem a comunicação de dados, agindo de forma que organize toda o conteúdo de informação em um específico Website. (MARQUES, 2019).

Atualmente o HTML é compreendido como um “esqueleto” de um site, possuindo a função de organização enquanto outras linguagens como CSS e JavaScript ganham destaque possuindo as funções principais que ditarão todo o layout do site.

Ainda de acordo com Marques (2019), a sua primeira versão foi baseada na SGML, uma linguagem de marcação utilizada para a estruturação de documentos, em suas versões sucessivas foram implementadas melhorias para que os programadores de fato utilizassem a linguagem como comum. Atualmente o HTML está em sua versão 5.0, com a utilização de suporte de áudio e vídeo em alto nível, sem a necessidade de uso de plug-ins, destacando principalmente que é uma versão que sempre ganhará avanços adiante, além de destacar a forte semântica estabelecida.

### 2.2.1.3 CSS

CSS é a sigla denominada para *Cascading Style Sheet* (Folha de Estilo em Cascatas), esta linguagem em específica é determinada por Linguagem de Estilo onde será utilizada para estilizar elementos escritos na linguagem de marcação o HTML. O CSS repara o conteúdo da representação visual do site, modificando quaisquer elementos depositados no site, como por exemplo ditar a cor específica de um texto e do fundo, sua fonte e espaçamento entre parágrafos. (GONÇALVES, 2022).

essa linguagem teve sua criação pelo W3C (*World Wide Web Consortium*) em 1996 com a finalidade de suprir aquilo que faltava no HTML, a formatação da página. Sua versão mais atualizada é de 2010, o CSS3 com a funcionalidade de melhorias de suas versões anteriores, a característica mais marcante dessa versão é a flexibilidade na criação de layouts, trazendo mais autonomia para os web designers e desenvolvedores. (...) O CSS pode ser compreendido em diferentes estilos, sendo em três tipos principais, o Estilo Interno onde CSS é carregado cada vez que um site é atualizado aumentando o tempo de carregamento, entretanto possuindo a facilidade de possuir todo conteúdo em apenas uma página para o compartilhamento do modelo para uma visualização, o Estilo Externo onde nesse estilo é abordado externamente em um arquivo .css, possuindo todo o estilo em um arquivo separado e aplicando a qualquer página desejada, além de diminuir o tempo de carregamento, e por último, o Estilo *Inline* trabalhado com elementos específicos da *tag* <style> onde cada elemento deve ser estilizado, ocorrendo de aumentar ou diminuir o tempo de carregamento, possuindo a utilidade de alterar apenas um elemento. (ID: MARQUES, 2019).

### 2.2.1.4 Conjunto de HTML, CSS e JavaScript

Segundo Ballerini (2021), o conjunto de três linguagens fazem a trindade principal de linguagens utilizadas para o Desenvolvimento Web, onde cada uma abriga suas funcionalidades que em conjunto formam uma criação de uma aplicação de qualidade.

Diante disso, podemos citar especificamente a composição de cada linguagem, estabelecendo uma dinâmica de formação do corpo humano de acordo com a função de cada uma delas, onde o HTML sendo uma Linguagem de Marcação responsável por estruturar os elementos da página, de forma simplificada e organizada para visualização, o CSS uma Linguagem de Estilos responsável por estilizar todos os elementos contidos na página, determinando seu layout e sua composição de cores de acordo com a preferência do usuário e o JavaScript uma Linguagem de Programação responsável por animar os elementos da página, podendo atualizar elementos dinamicamente e lidar de maneira melhorada o envio/recebimento de dados na página.

### 2.2.2 Back-end

Back-end é a área da programação voltada para parte lógica de sites, aplicativos, softwares ou outros tipos de sistemas de informação, focando na funcionalidade de integração, isso acaba garantindo seu bom funcionamento e seus objetivos de acordo com a necessidade. Normalmente desenvolvedores de ambas as áreas front-end e back-end acabam trabalhando juntos para a conclusão de sua infraestrutura. (ROVEDA, 2021) e (MOLINA, 2020).

os profissionais deste campo são responsáveis por ditar os comandos de uma determinada aplicação de forma que opere de acordo com objetivo estabelecido, utilizando as principais Linguagens de Programação para execução da aplicação em diferentes ambientes. (LONGEN, 2022) e (Back-end: O Que É, Para Que Serve e Quais Suas Linguagens, 2021)

Segundo Roveda (2022) sendo as principais linguagens de programação, dentre delas estão PHP, Ruby, Python, Java e Swift. Além utilizar alguns frameworks, dentre deles estão React Native, Xamarin, Swiftic, Ruby on Rails, ExpressJS e Laravel.

### 2.2.3 Desktop

A Desktop é uma plataforma de computadores parte da interface gráfica onde exibe a principal área dos sistemas operacionais, permitindo o indivíduo ter acesso aos elementos que o faz parte dela, por exemplo pastas, programas, entre outros. ela pode ser instalada em qualquer computador para executar as tarefas desejadas.

Segundo Desktop (2022), esta plataforma proporciona benefícios à rotina do indivíduo, dispondo uma ótima estrutura para se concentrar nas atividades que deseja desenvolver sem nenhum impedimento.

**2.2.4 Frameworks**

framework são estratégias que visam solucionar problemas específicos. Basicamente, são bibliotecas de códigos prontos que constituí uma funcionalidade específica, incorpora em sistemas agilizando a codificação de tais partes. As suas funções determinam as tarefas repetitivas e simples, que aparecem comumente em softwares. Assim, o desenvolvedor reutiliza esses códigos. Essa funcionalidade é como um automatizado que faz parte do trabalho do desenvolvedor para que ele foque em outros assuntos. (...) os frameworks são criados por comunidades de desenvolvedores que orientam a consistência e rigorosidade, modificando e revisando o código para um padrão de qualidade, um dos exemplos dessa qualidade seria a legibilidade da construção do código, um código limpo é importante para os desenvolvedores para que os desenvolvedores realizem as manutenções e entender do que se trata ao ler o código. (...) é a economia de tempo, pois eles são os que facilitam no desenvolvimento dos sistemas, assim não perdendo tempo com funcionalidades básicas, podendo progressivamente aumentar a produtividade, fazendo jus a resultados satisfatórios aos clientes. (NOLETO, 2020)

Segundo Clark (2020), alguns Frameworks utilizados demonstradas, para o desenvolvimento são:

* React Native;
* Xamarin;
* Swiftic;
* Ruby On Rails;
* ExpressJS;
* Laravel.

### 2.2.3.1 React Native

Segundo Borah (2019), React Native foi introduzido pelo Facebook em 2018, como framework JavaScript para criar aplicativos nativos para qualquer plataforma e dispositivo. Pode-se desenvolver aplicativos para ambas as plataformas Android e iOS, além das criações de versões específicas da plataforma de vários componentes, permitindo o fácil uso da base de código em plataformas múltiplas.

Destacando recursos como o desenvolvimento *low-code* (criação de software por meio de UI), a compatibilidade com plugins de terceiros e API declarativa para IU preditiva.

### 2.2.3.2 Xamarin

Xamarin foi introduzido pela Microsoft, é um framework baseada em .NET para a criação de aplicativos de plataformas cruzadas e código aberto, ele oferece um espaço para o desenvolvimento em Back-end, API, entre outros. Ela é suportada por várias ferramentas de trabalho, bibliotecas e linguagens de programação. Se cria aplicativos nativos com suporta a várias plataformas, incluindo o Android e iOS. (BORAH, 2019)

Ainda de acordo com Borah (2019), alguns recursos que se destacam são:

* Grande comunidade;
* Infraestrutura versável do Back-end;
* Carregador de aplicativos;
* Gerenciador de SDKs Android;
* Gerenciador de emulador do Google.

### 2.2.3.3 Swiftic

Swiftic foi introduzido pela Conduit, é um framework de fácil uso para o desenvolvimento iOS sem o uso de códigos, apenas uma interface intuitiva sendo possível criar, administrar e publicar no painel de controle. Suas características são criar notificações, comprar, assistências, entre outros. (BORAH, 2019).

### 2.2.3.4 Ruby on Rails

Ruby on Rails é um framework de código aberto escrito em Ruby, seguindo o padrão arquitetural MVC (Model-View-Controller) que, recebendo uma requisição, solicita ao Model informações necessárias, que quando retornam ao Controller, as envia para a View que renderizará o Model e a parte loja da aplicação, a qual gerencia o comportamento de dados. Sendo assim, os recursos do banco de dados, as notificações, entre outros. Ele tem o necessário para que tudo se faça, mas não sabe quando irá executar em conjunto da View gerencia a saída gráfica e textual da aplicação visível ao usuário. Ela não possui a responsabilidade de saber quando exibir os dados, apenas como exibi-los e o Controller é a chamada que chamará, quando chamar, executará uma determinada ação, interpreta a entrada do mouse e teclado do usuário, conduzindo o modelo a se alterar em uma forma propícia. (ANDRADE, 2020).

Ainda de acordo com Andrade (2020), a vantagem de utilizar é dada pela multiplataforma, open-source, ter uma comunidade ativa, a facilidade no desenvolvimento de sites orientados a banco de dados, ter um rico ecossistema com uma variedade de bibliotecas externas, possui também ferramentas para desenvolver, como o Extreme Programming e Scrum.

### 2.2.3.5 Express.js

O Express.js é um framework para Node.js. Ele foi projetado para construir página única, página múltipla e aplicações de web híbridas, como um esqueleto, se usa aplicativos baseados em servidor(es) que recebe solicitações de entrada dos usuários. Ele é construído a partir da configuração e simplicidade do middleware Connect, permitindo várias extensões e recursos úteis. (SHAZAD, 2016) e (ENE, 2017).

Segundo HolowayChuk, Caswell e Wilson (2022), um dos exemplos é o Connect, um pequeno framework para node.js usando plugins chamado *middleware*.

Middleware é um software que fornece serviço e recursos para aplicativos, são funções lidando com solicitações). O Connect envolve objetos dos módulos HTTP de node.js, como Server, ServerRequest e ServerResponse, dando-os novas características, um dos quais permitem o Server usar uma quantidade de middleware. (SUGDEN, 2011).

Segundo Mardan (2013), o objetivo do framework é não precisar repetir códigos. Node.js, sendo um mecanismo E/S na qual possui módulo HTTP, se usando esse módulo, grandes trabalhos como a análise de pacotes, cookies, sessões de armazenamento (tanto na memória, como em Redis), entrarão para a seleção do padrão de rota correto com expressões reguladas assim sendo reimplementado. Então, no caso, Express.js estaria ali para seu uso.

Baseado nas palavras de Zivadinovic (2015), Mardan (2013) e Ettinger (2012), o Redis é um armazenador de chave-valor conhecido pela sua velocidade de performance para armazenar sessões juntamente com o Express.js, porém não é necessário, o Express.js tem armazenamento de memória para sessões. Redis pode também ser usado para enfileirar trabalhos.

### 2.2.3.6 Laravel

Laravel é um framework que é desenvolvido em PHP por Taylor B. Otwell, sua utilização poupa o tempo e os custos para quem o utiliza, pois ele é um conjunto de bibliotecas que cria uma base onde se constroem e otimizam recursos nas aplicações. (ANDRADE, 2019).

Ainda de acordo com Andrade (2019), seu objetivo seria resolver os problemas que têm atuações genéricas, isso permite ao desenvolvedor focar nos bugs e erros da aplicação, não na arquitetura.

Os recursos do Laravel é o sistema de template (Blade), módulo de autenticação e o padrão MVC. O sistema de *template* facilita a criação da camada de visualização dos dados da página HTML. Com ele, pode-se criar páginas simples e intuitivas rapidamente e eficaz. Com os recursos do Blade se destacam: a Herança de Layouts, o sistema de tags, as seções e uso de código PHP nos templates.

a herança de layouts centraliza o layout padrão em um ou mais arquivos, ele estende arquivos, aproveitando o layout de diferentes visualizações, minimizando a quantidade de digitação do código. Com o sistema de tags, pode-se usar templates na sintaxe fluente do estilo Laravel, usando configurações de parâmetros explícitos, pegando dados contextuais, buscar a saída de um loop e paginação. (FERREIRA, 2018).

Segundo Andrade (2019), as seções injetam o layout de conteúdo pai de um layout extendido do blade filho. O módulo de autenticação completa é o que provê todos os recursos para a sua implementação, como o cadastro, a autenticação e autorização de usuários, recuperação de senhas, login logout das sessões.

Laravel não precisa criar um código SQL para manipular uma tabela no banco de dados, todo o processo é feito com o código PHP, sendo convertido em instruções SQL. Ele implementa o padrão Active Record, onde o model da aplicação representa a tabela de um BD.

### 2.2.4 Plataformas Mobile

As plataformas Mobile (ou também sistemas operacionais mobile) são aqueles que permitem o usuário rodar softwares e serviços desejáveis no seu dispositivo móvel, também fornecem ferramentas que permite aplicativos compartilharem dados e serviços. Mobile é a parte da interface gráfica que permite softwares e serviços (como por exemplo a câmera, chamadas e geolocalização) a serem transmitidos em um dispositivo móvel.

Algumas plataformas mais utilizadas são iOS, Android e HarmonyOS.

### 2.2.4.1 Android

Segundo o site TechTudo e os autores Kadambari, Han, Tilak e Houng (2012), é uma plataforma da Google inicialmente desenvolvida pela Android Inc, sendo introduzida nos tele móveis em 2008 nos EUA presente em múltiplas fabricantes de dispositivos, como a Samsung, Motorola e LG. Baseado no núcleo Linux, sendo um código aberto e tendo acesso a personalização da plataforma. Por esses motivos, o Android não será igual em todos os dispositivos por conta de que empresas podem personalizar sua interface.

### 2.2.4.2 iOS

Segundo Magno (2017), iOS é um sistema operacional móvel da Apple Inc. desenvolvida nos EUA presente apenas nos dispositivos da Apple, ela se consiste em quatro camadas, Core OS, Core Services, Media e Cocoa Touch.

A camada Core OS contém características de baixo nível utilizada para implementar outras tecnologias, onde o desenvolvedor lidera com a segurança ou a comunicação, ele pode fazer isso usando frameworks.

A camada Core Services contém os serviços fundamentais do sistema, onde, mesmo que o desenvolvedor não utilize os serviços diretamente, o sistema construirá em cima disso, suas tecnologias disponíveis são: *Grand Central Dispatch*, *In-app Purchase*, *SQLite* e *XML Support*.

Grand Central Dispatch é o que o desenvolvedor utiliza para gerenciar as tarefas executadas no aplicativo, otimizando e oferecendo uma threading (divisão de processos entre duas ou mais tarefas que se executam concorrencialmente) eficiente. Ele também proporciona vários tipos de tarefas de baixo nível, como por exemplo, monitorar eventos do processo e escrever em arquivos. (MAGNO, 2017).

Ainda de acordo com Magno (2017), In-app Purchase é usada para que os desenvolvedores consigam vender seus serviços dentro dos aplicativos.

SQLite incorpora um banco de dados SQL leve, não tendo que executar largos processos de banco de dados remoto. O desenvolvedor pode criar bancos de dados locais e gerenciar os registros a partir de seu aplicativo.

XMLSupport oferece suporte adicional para a manipulação .XML, podendo também transformá-lo para HTML.

A camada Media são as quais contém as tecnologias gráficas, visuais e sonoras, essa camada foi feita para a fácil manipulação à criação de aplicativos multimídias. Frameworks podem acessar essas tecnologias e ferramentas que o desenvolvedor possa utilizar para criar aplicativos de maneira mais complexa. (MAGNO, 2017).

De acordo com Kadambari, Han, Tilak e Huong (2012) e Magno (2017), a camada Cocoa Touch, contém os frameworks principais para a construção de aplicativos dessa camada, ela define a infraestrutura para tecnologias fundamentais, como multitarefa, o serviço de notificação e os serviços de alto nível do sistema

### 2.2.4.3 Harmony OS

Harmony OS é um sistema operacional móvel da Huawei desenvolvida na China lançado em 2019 e é projetado com camadas assim como o iOS, as funções de seu sistema são expandidas por níveis, de sistema para subsistemas, além dos módulos (HARMONYOS DEVELOPER, 2021) e (LI, 2021)

Segundo o site HarmonyOS Developer (2021), a camada kernel e seu primeiro subsistema, o subsistema Kernel. HarmonyOS projeta multi-kernel para seus dispositivos com diferentes limitações de recursos, assim ele é selecionado para que os kernels sejam apropriados ao sistema. Seu segundo subsistema, o subsistema Driver estabelece bases para o ecossistema do hardware do HarmonyOS. O desenvolvimento e o gerenciamento dos drivers são fornecidos pelo acesso unificado dos dispositivos.

A camada de serviço do sistema, e, um exemplo de subsistema, na qual é o Conjunto de subsistemas de recursos básicos do sistema, onde esse conjunto fornece recursos básicos como gerenciamento de dados, agendador, gráficos, segurança, inteligência artificial, entre eles, também se contém o runtime Ark, onde usa linguagens para fornecer bibliotecas de classe do sistema. A camada do framework é aquela usada para fazer aplicativos para uma variedade de linguagens, também com APIs.

De acordo com o site HarmonyOS Developer (2021), a camada da aplicação, onde se consiste nos aplicativos do sistema e de terceiros. Cada aplicação de HarmonyOS são sustentadas pelas Habilidades de Recursos e Partículas. A Habilidade de Recurso dá as interações do usuário pela interface do usuário. A Habilidade de Partículas oferece o processamento das tarefas em segundo plano e o acesso dos dados.

## Paradigmas de Programação

paradigmas de Programação são estilos de programação nos quais os profissionais se baseiam para solucionar um determinado problema, especificamente são determinados como metodologias. Ao contrário das Linguagens de Programação, os paradigmas se trata de forma como o profissional irá recorrer a solução de um problema usando uma determinada linguagem de programação. (...) as linguagens de programação operam em cima dos Paradigmas, este que seguiram como caminhos para determinar a solução de um problema, dessa forma, as linguagens seguiram caminhos diferentes dependendo de suas características, podendo se enquadrar em apenas um paradigma ou multiparadigma, assentando-se segundo a fonte. (NOLETO, 2020)

Ainda de acordo com Noleto (2020), os Paradigmas são classificados em alguns tipos como Imperativo, Declarativo, Funcional, Lógico, Orientado a Objetos e Orientado a Eventos.

### 2.3.1 Paradigma Imperativo

o paradigma Imperativo denominado também como Paradigma Procedural, neste paradigma o profissional deve comandas as instruções ao computador na sequência em que deverão ser executadas, um exemplo de linguagens que entram neste tipo de paradigma são como Cobol, Fortran e Pascal. Este paradigma é recomendado a projetos que não irão possuir muitas modificações ao longo de seu desenvolvimento, pois estabelecem muita dependência das instruções, caso alguma instrução não seja estabelecida de forma correta a solução do problema não acontecerá de forma esperada pelo cliente. (NOLETO, 2020).

Ainda de acordo com Noleto (2020), algumas linguagens que abordam este paradigma são as linguagens C, C++, Java e Pascal.

### 2.3.2 Paradigma Declarativo

o Paradigma declarativo é responsável por dar mais foco ao que deve ser resolvido, dando foco principal ao problema sugerido do que necessariamente a sua resolução. Neste paradigma é obtido o nível de abstração, onde o profissional deverá buscar o máximo de detalhes em sua solução, possuindo maior sequência lógica e qual o resultado esperado. (NOLETO, 2020)

Ainda de acordo com Noleto (2020), possuindo principalmente instruções lógicas imutáveis, onde serviram de interação no qual gerará o resultado esperado, havendo o mesmo resultado independente das modificações ocorridas durante o processo.

Algumas linguagens que abordam este paradigma são Absys, Ciao e Alice.

### 2.3.3 Paradigma Funcional

o Paradigma Funcional é responsável por ditar as instruções em cima de funções, onde o problema será dividido em blocos destacando para sua resolução com o uso da implementação de funções definindo variáveis em seu escopo para que retornem algum resultado. (NOLETO, 2020).

Conforme Noleto (2020) afirma, este paradigma é recomendado quando a solução necessita de resultados dependentes de base matemática, dessa forma as funções implementadas farão os cálculos matemáticos, para que não haja erros no resultado, elas possuem alocação de memória automática, eliminando duplicação de dados ao realizar outra operação em diferentes momentos. Algumas linguagens que abordam este paradigma são Haskell, Scala, Racket e JavaScript.

### 2.3.4 Paradigma Lógico

O Paradigma Lógico é responsável principalmente da lógica do profissional responsável, derivando do Paradigma declarativo. Utilizando formas de lógica simbólica como padrões de entrada e saída, realizando inferências que demonstrem os resultados. (NOLETO, 2020).

Ainda de acordo com Noleto (2020), este paradigma é implementado principalmente na solução de problemas envolvendo inteligência artificial, criação de programas especialistas e comprovação de teoremas. Algumas linguagens que abordam este paradigma são QLISP, Mercury e Prolog.

### 2.3.5 Paradigma Orientado a Objetos

O Paradigma Orientado a Objetos ou POO, é responsável por produzir programas mais rápidos com maior confiabilidade e a custo menor, sua popularização ganhou forma por apresentar uma das linguagens mais populares, Java, ao permitir uma programação multiplataforma de uma mesma maneira. (NOLETO, 2020).

Ainda de acordo com Noleto (2020), este paradigma surgiu como proposta para solução de problemas na indústria de software, apoiando-se em características de classe e objeto retratando a programação assim como vista no mundo real.

Uma linguagem de programação ao ser classificada ao uso do paradigma orientado a objetos deve implementar as três principais características de seus conceitos, a Herança, o Polimorfismo, o Encapsulamento e a Abstração. Algumas linguagens que abordam este paradigma são Java, C++, C# e Python.

### 2.3.6 Paradigma Orientado a Eventos

O Paradigma Orientado a Eventos é utilizado pelas linguagens de programação que utilizam recursos gráficos, principalmente no desenvolvimento de jogos e formulários. Neste paradigma é utilizado a cláusula de dependência de um evento para o desenvolvimento do projeto que são disparados pelo usuário, dessa forma, quem utiliza é responsável pelo momento que o programa será executado. (NOLETO, 2020).

## 2.4 Engenharia de Software

De acordo com Noleto (2020), podemos dizer que a engenharia de software é um âmbito da ciência da computação que abarca a estrutura e desenvolvimento de Aplicativos e Softwares. Esta que opera de acordo com princípios, métodos e práticas que foram diligentemente moldados através do tempo, aperfeiçoando-se o conceito de software e tecnologia.

o intento da engenharia de software é ministrar um produto – seja ele um app, site ou qualquer aplicação desenvolvida. Uns dos seus princípios são que acaba sendo confiável, eficiente e eficaz em seu desígnio primário. Se tais requisitos não se cumprirem, o produto não sucede do estágio de produção. Tal veniaga é reiteradamente arquitetado para aplicar-se em máquinas reais, utilizadas na realidade, em situações reais. (NOLETO, 2020).

Ainda de acordo com Noleto (2020), as engenharias de software principais são UML, Diagrama de Classes e Casos de Uso.

### 2.4.1 UML

A sigla UML, vinda do nome Unified Modeling Language (Linguagem Unificada de Modelagem), se trata de uma linguagem de notação utilizada para modelar e documentar as diversas fases do desenvolvimento de sistemas orientados a objetos. Para auferir tal desígnio ele prevê uma série de elementos gráficos – como formas geométricas de retângulos, linhas, setas e balões – dos quais empenham o papel de espelhar componentes de um aplicativo, suas interações e mudanças. (NOLETO, 2020).

Ainda de acordo com Noleto (2020), ao se arquitetar uma aplicação, se premedita dezenas de funções, relações e requisitos para tal; o diagrama UML se assenta no quesito de evadir os erros das fases de especificação do projeto e na comunicação entre as diferentes partes do programa, como gerentes e pessoas desenvolvedoras, por exemplo. Esse esboçamento do software permite uma intelecção do sistema, deixando claro todas as entidades e visionando os erros de implementação.

Há profusos tipos de diagramas UML e eles se diferem em duas castas, segundo Noleto (2020), subdivididos em diagramas estruturais e comportamentais. Cada um deles é usado para especificar, documentar, modelar e visualizar aspectos específicos de uma aplicação. A matriz de diagramas é grande – como por exemplo em diagramas estruturais: diagrama de classe, diagrama de objetos, diagrama de componentes, digrama de implantação, diagrama de pacotes, diagrama de estrutura e em diagramas comportamentais: diagrama de caso de uso, diagrama de máquina de estados, diagrama de atividades e diagrama de interação – dos quais serão especificados em um momento ulterior.

### 2.4.2 Diagrama de Classes

Segundo o site, Lucidchar (2020), os Diagramas de Classes são divididos em duas categorias, Diagramas Estruturais e Diagramas Comportamentais.

### 2.4.3 Diagramas Estruturais

De acordo com o site LucidChart (2020), todos são logrados para modelar os aspectos estáticos do sistema, como a arquitetura, classes, interfaces, métodos e componentes.

1. Diagrama de classes

Este é um dos diagramas mais apaniguados para o processo de desenvolvimento de software e serve como diagrama primigênio para a criação de outros diagramas. Tal importância advém de sua praticabilidade no quesito de mapear o sistema por meio da modelagem dos seus métodos, atributos e classes – além dos inter-relacionamentos, como herança, composição, associação e dependência. (LUCIDCHART, 2020).

1. Diagrama de objetos

Este diagrama é empregado para visualizar instâncias específicas das classes que foram definidas no diagrama de classes. Nessa ótica, ele mostra os abjetos e seus valores e relacionamento, em determinado momento da execução do programa. (LUCIDCHART, 2020).

1. Diagrama de estrutura

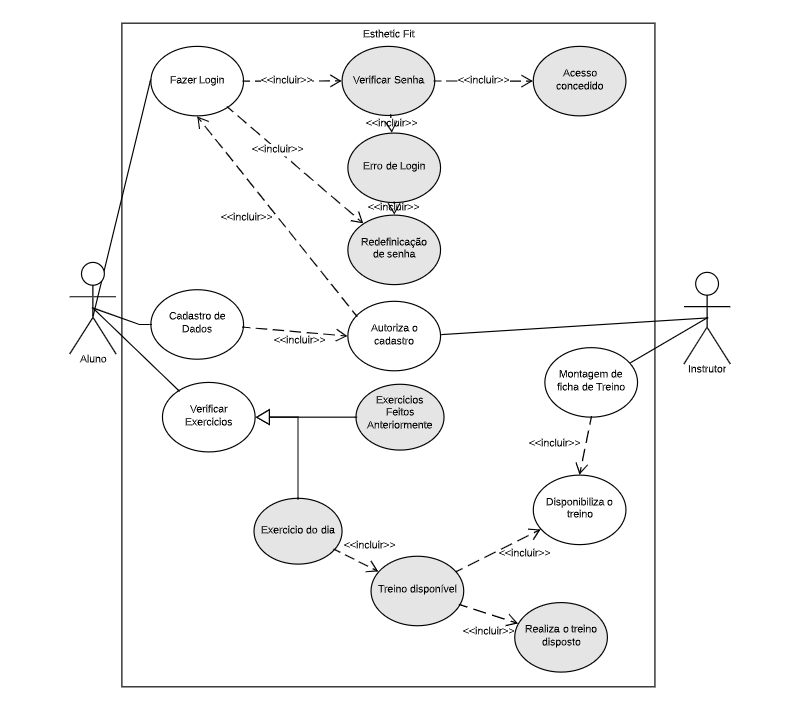
Através deste diagrama é possível retratar as colaborações internas que ocorrem entre classes, componentes, instâncias ou interfaces para a execução de uma tarefa específica. (LUCIDCHART, 2020).

### 2.4.4 Diagramas Comportamentais

1. Diagrama de casos de uso

Aplicado precipuamente na fase de especificação dos requisitos, este diagrama documenta as funcionalidades do sistema e as interações do usuário com cada uma. Se trata de um diagrama simples e não requer grande conhecimento para operá-lo. (LUCIDCHART, 2020).

Figura 6 - Diagrama de Caso de Uso UML



Fonte: Imagem dos Autores, 2022.

1. Diagrama de atividade

No diagrama de atividade é salientado o fluxo percorrido até a conclusão de uma determinada atividade. Mediante dele, pode-se exprimir as funções feitas entre os abjetos durante cada função (LUCIDCHART, 2020).

1. Diagrama de máquina de estados

Neste diagrama é descrito como uma aplicação correspondente aos eventos internos e externos. É possibilitado significar as mudanças de estado que um objeto apresenta durante a execução de processos (LUCIDCHART, 2020).

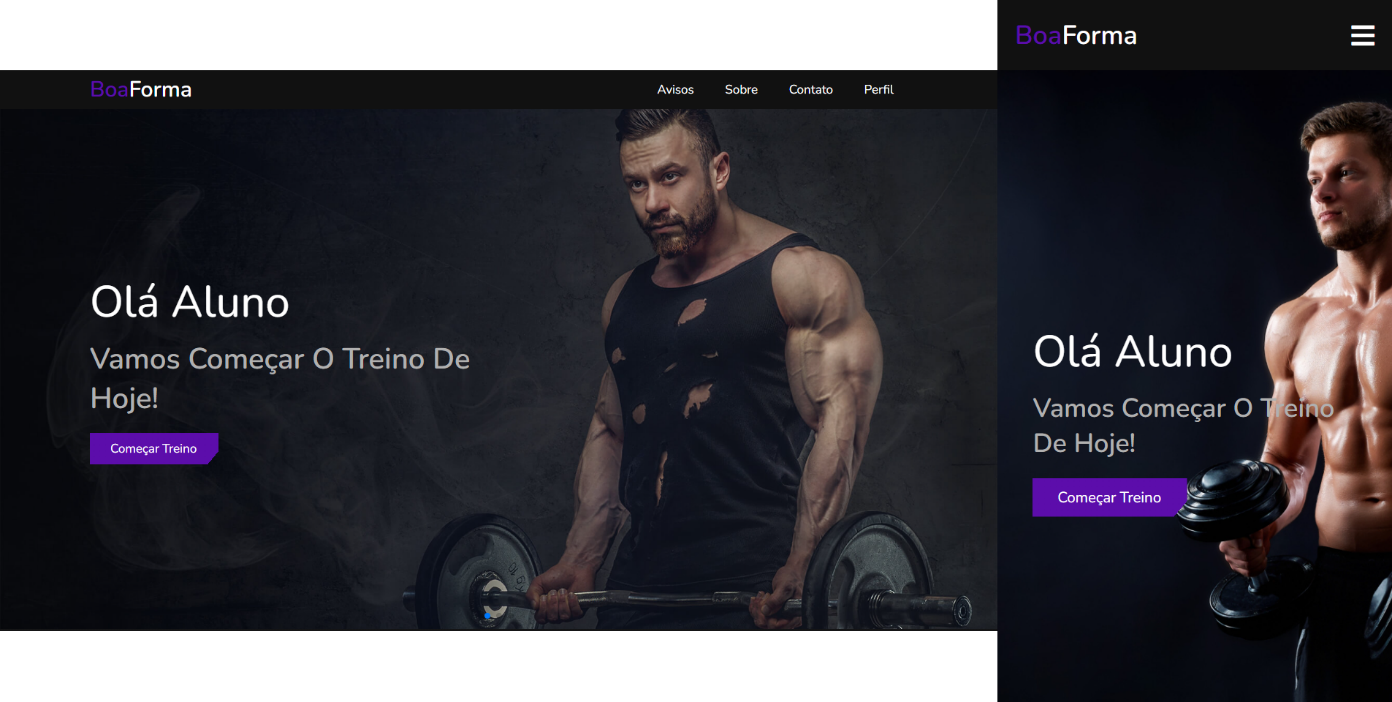
## 3 DESENVOLVIMENTO

A criação do software responsivo para a academia “Boa Forma” localizada em Santo Antônio de Posse, São Paulo, obtém como funcionalidade otimizar o sistema atual contando com a melhoria da comunicação entre Professores e Alunos do ambiente.

Para resolução contra o problema do cenário pandêmico enfrentado pelo estabelecimento nos anos de 2020 a 2021, destacando-se principalmente a utilização de métodos antigos que abordam contra as medidas preventivas, como a utilização de fichas de cadastro e manipulação de dados, dessa forma, com a otimização do sistema de cadastro, manipulação de dados, avisos, contato e listagem de exercícios optará por metodologias tecnológicas que abordem todas as necessidades solicitadas pela Academia Boa Forma, adotando metidas preventivas e melhoria de comunicação para o estabelecimento atual, auxiliando para o combate do problema.

A seguir os protótipos do software responsivo, sendo esclarecidos de acordo com os requisitos estabelecidos pela gerência da Academia Boa Forma. Na Figura 7 é demonstrado a Home Screen.

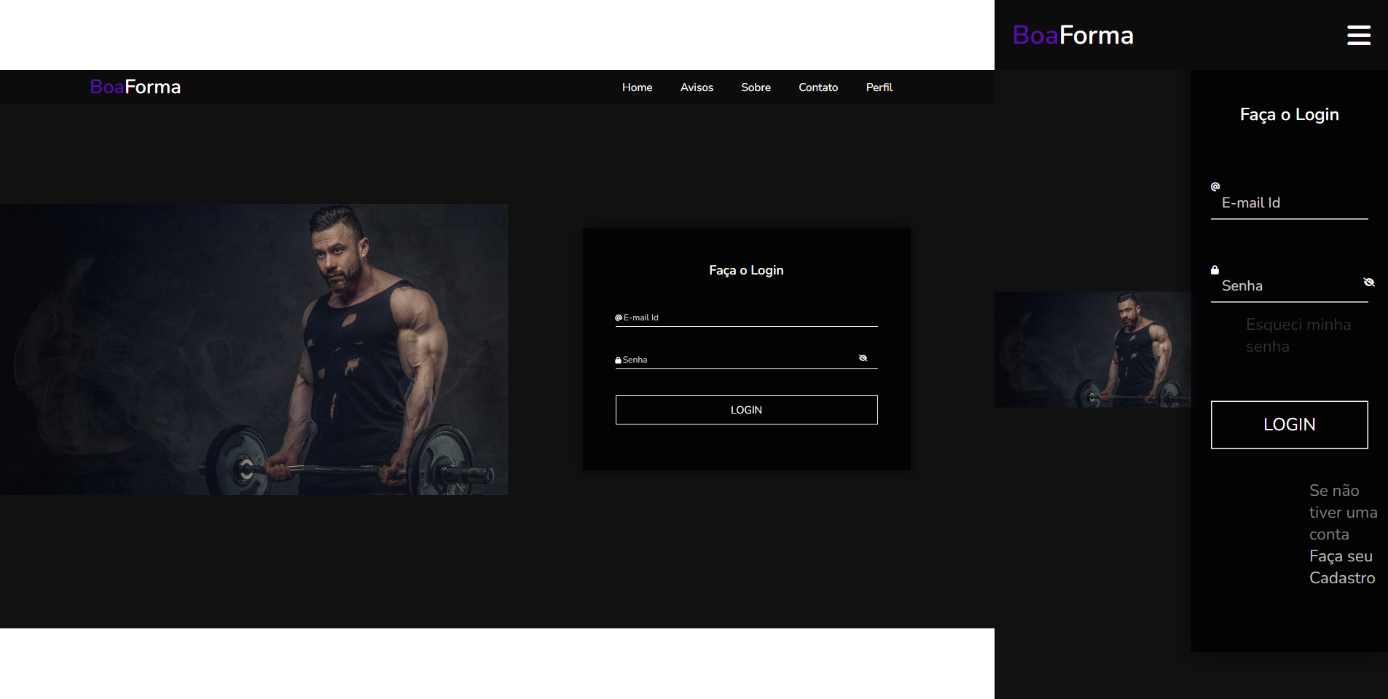
Figura 7 - Home Screen



Fonte: Imagem dos Autores, 2022.

* Home Screen: Tela de Início do Usuário, com a recepção da primeira vista do usuário com o software responsivo. Primeira tela onde rediciona os usuários as outras telas sendo, Tela de Login ou Tela de Cadastro. A seguir na Figura 8 é obtida a Login Screen

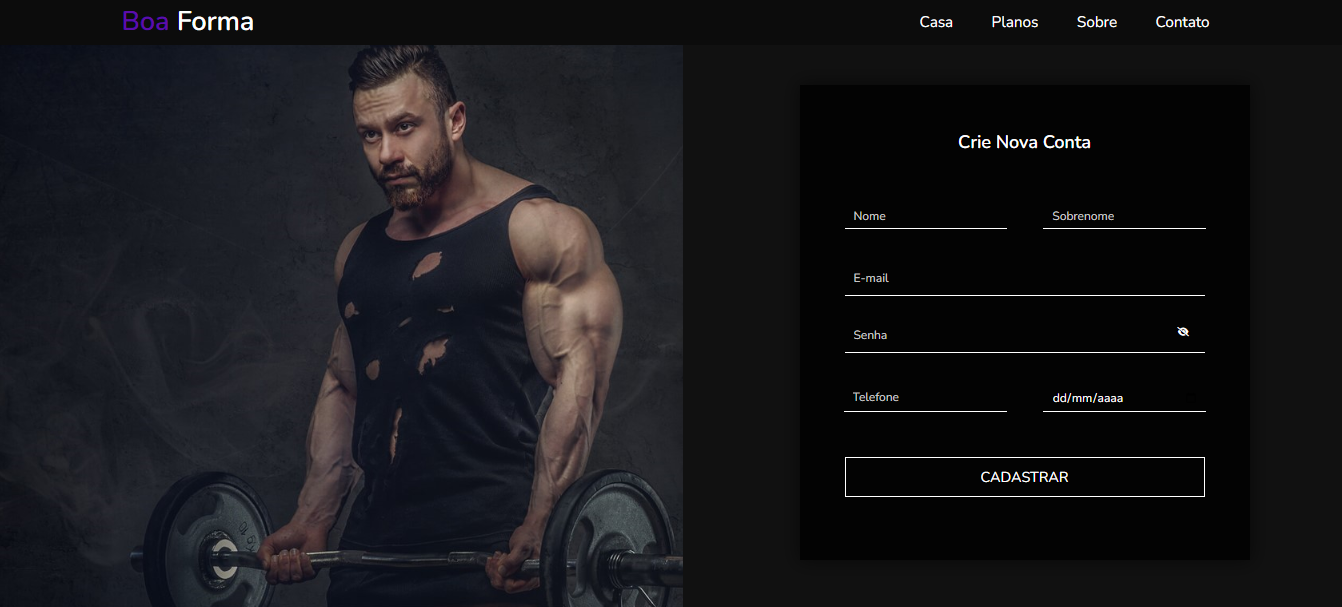
Figura 8 - Login Screen



Fonte: Imagem dos Autores, 2022.

* Login Screen: Tela de Login, com a funcionalidade de logar o usuário para o acesso de seus treinos ou restante das informações do site, nesse caso, o usuário já deverá conter seus dados cadastrados;
* Caso o usuário não tenha seus dados cadastrados pode optar para cadastrar os mesmos com a New Account Screen, sendo redirecionado para está baixa por meio do Link - Faça seu Cadastro ao pé do Formulário de Login. Na Figura 9 a demonstração da New Account Screen.

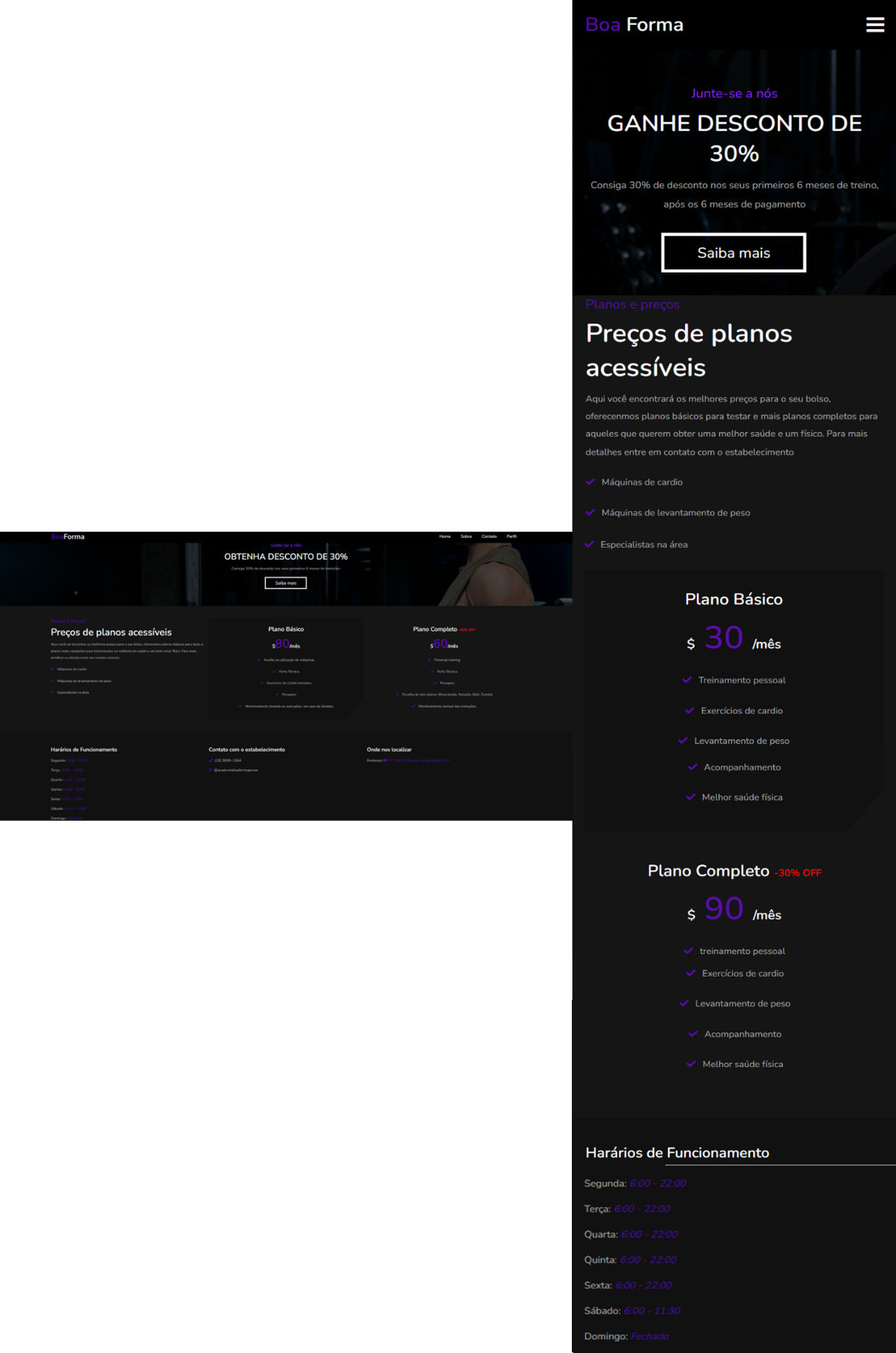
Figura 9 - New Account Screen



Fonte: Imagem dos Autores, 2022.

* New Account Screen: Tela de Cadastro, com a funcionalidade de cadastrar os dados pessoais do usuário para a manutenção de seus planos, estabelecendo a criação de métodos viáveis em cima de seus objetivos.
* Após cadastrar seus dados o usuário será redirecionado para a Training Screen para o acesso de seus treinos diários. Na Figura 10 é relatado a Planes Screen.

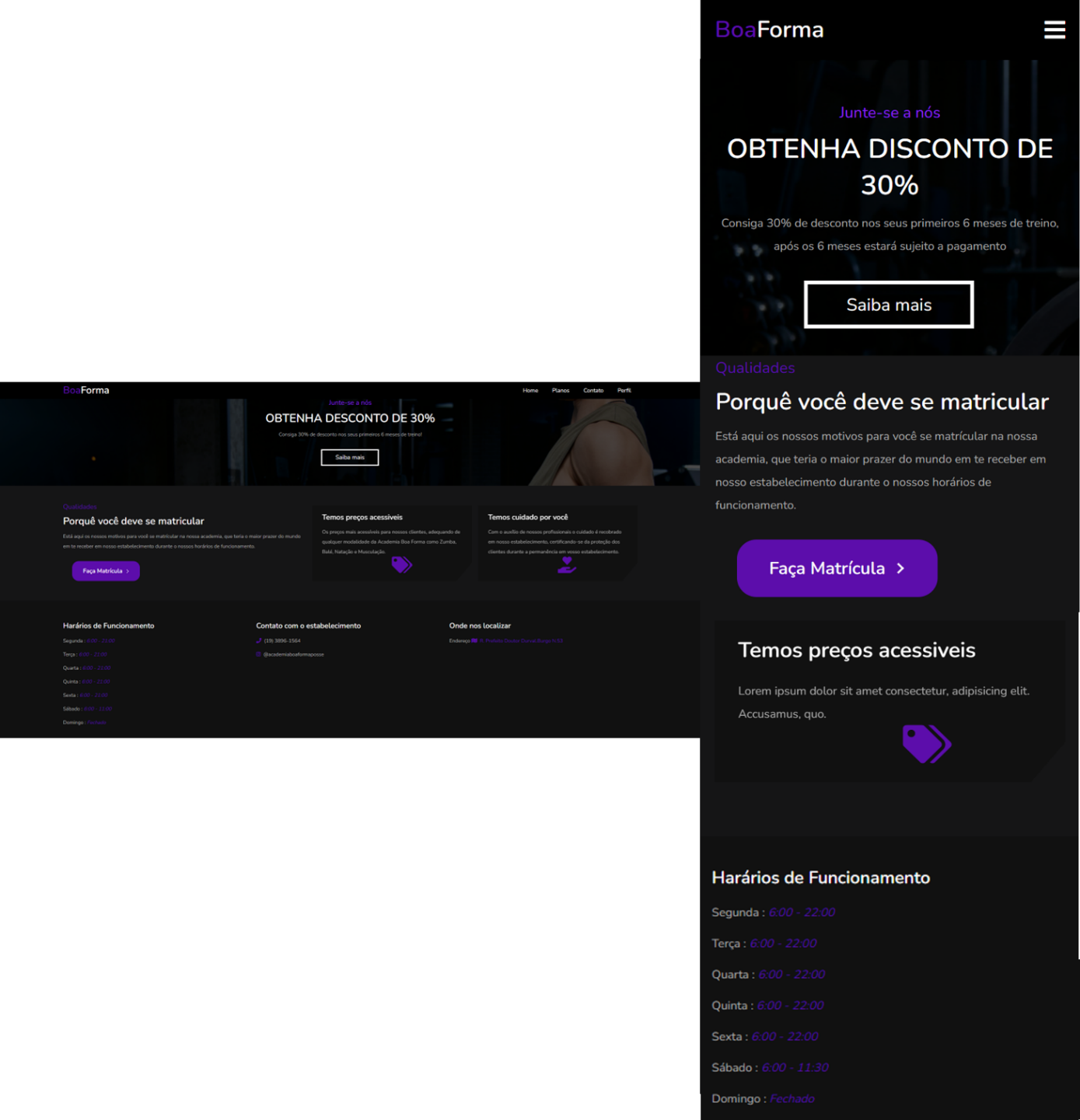
Figura 10 - Planes Screen



Fonte: Imagem dos Autores, 2022.

* Planes Screen: Tela de Planos, com a funcionalidade de demonstrar ao usuário os planos da Academia Boa Forma, onde serão estabelecidos do Plano Básico ao Plano Completo, juntamente com valores e promoções diárias;
* Nesta página, o usuário poderá consultar sobres os horários de funcionamento, Endereço do estabelecimento juntamente com o contato, podendo ser realizado através do Número ou das Redes Sociais (Instagram). A Figura 11 é relatado a Info Screen.

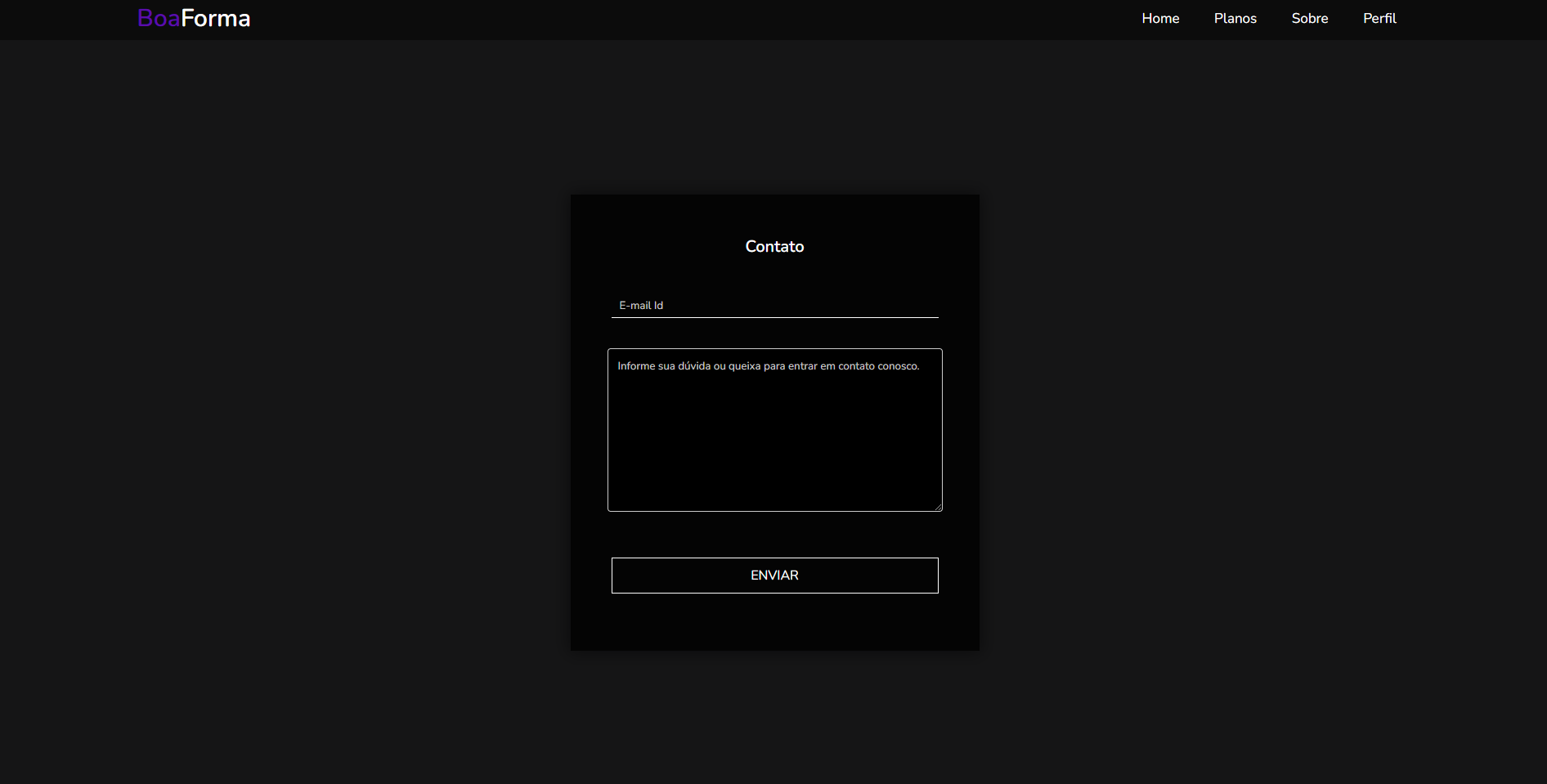
Figura 11 - Protótipo Info Screen

**

Fonte: Imagem dos Autores, 2022.

* Info Screen: Tela de Informações, com a funcionalidade de entreter o usuário com justificativas de por que utilizar a Academia Boa Forma;
* Nesta tela consta outras informações como os horários de Funcionamento, a localização e contato, caso necessário ou em via de dúvidas sobre a academia, sendo realizado através do Número ou das Redes Sociais (Instagram);
* Servindo como justificativas sendo: Preços Acessíveis, Cuidado, Atendimento Personalizado e Busca do alcance dos objetivos estabelecidos;
* Caso o usuário não contenha uma matrícula, poderá se cadastrar no sistema pelo Botão “Faça Matrícula”, sendo redirecionado a New Account Screen para realizar seu cadastro e estar dentro dos planos e preços da Academia Boa Forma;
* Caso o usuário queira saber mais sobre os descontos, poderá consultar a Planes Screen pelo Botão “Saiba Mais”, sendo redirecionado para a consulta de Planos Acessíveis e Descontos. A Figura 12 é relatada a Contact Screen.

Figura 12 - Protótipo Contact Screen



Fonte: Imagem dos Autores, 2022.

* Contact Screen: Tela de Contato, com a funcionalidade de atender dúvidas ou reclamações a Contact Screen estabelece um vínculo com o usuário para atender ou melhorar qualquer aspecto que atrapalhe sua convivência no estabelecimento da Academia Boa Forma.

## CONCLUSÃO

Diante da pesquisa realizada pretendeu-se entender como a problematização do contato físico por conta da transmissão do vírus COVID no período de março de 2020 até os dias atuais, dos quais influenciou na manutenção de sistemas manuais de cadastro e funcionamento de alguns estabelecimentos, principalmente os estabelecimentos que utilizam maior contato físico, como exemplo a Academia Boa Forma em Santo Antônio de Posse (SP). Nesse sentido, foi desenvolvido um software responsivo que auxiliasse na manutenção de cadastro e funcionamento dos dados da academia local, otimizando o sistema manual para um sistema virtual, auxiliando a melhora de comunicação entre os clientes e os profissionais da academia, fazendo-se o uso de tecnologias de desenvolvimento como Linguagens de Programação, Plataformas de Desenvolvimento, Paradigmas de Programação e Engenharia de Softwares. Em conjunto para o desenvolvimento foram utilizados

# REFERÊNCIAS

About HarmonyOS. **Developer HarmonyOS**, 25 de julho de 2022. Disponível em: <https://developer.harmonyos.com/en/docs/documentation/doc-guides/harmonyos-overview-0000000000011903>. Acesso em: 16 jun. 2022.

ANDRADE, A. P. O que é Laravel*?* **TreinaWeb**, 16 de maio de 2019. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-laravel>. Acesso em: 11 set. 2022.

ANDRADE, A. P. O que é a Linguagem de Programação Ruby*?* **Treina Web**, 25 de agosto de 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-a-linguagem-de-programacao-ruby>. Acesso em: 9 mai. 2022.

ANDRADE, A. P. 31 de agosto de 2020. O que é Ruby on Rails*?* **TreinaWeb**, 31 de agosto de 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-ruby-on-rails>. Acesso em: 9 mai. 2022.

APRENDA PHP do Básico ao Profissional. [S. l.], 2022. Disponível em: <https://www.devmedia.com.br/php/>. Acesso em: 11 set. 2022

Back-end: O Que É, Para Que Serve e Quais Suas Linguagens? **Ewally**, 6 de setembro de 2021. Disponível em: <https://www.ewally.com.br/blog/ajudando-sua-empresa/backend/>. Acesso em: 9 mai. 2022.

BALLERINI, R. HTML, CSS e Javascript, quais as diferenças? **Alura**, 25 de fevereiro de 2021. Disponível em: <https://www.alura.com.br/artigos/html-css-e-js-definicoes>. Acesso em: 9 mai. 2022.

BEDRAN, A. F. Swift: conheça a linguagem de programação do iOS. **Zup**, 21 de julho de 2021.Disponível em: <https://www.zup.com.br/blog/swift-linguagem-de-programacao-ios>. Acesso em: 9 mai. 2022

BORAH, R. Top Mobile App Development Frameworks in 2022. **Clarion Technologies**, 23 de agosto de 2019.Disponível em: <https://www.clariontech.com/blog/top-mobile-app-development-frameworks-in-2019>. Acesso em: 9 mai. 2022.

BRATFISCH, G. Ruby: para que serve essa linguagem de programação? **HostGator**, 9 de junho de 2021*.* Disponível em: <https://www.hostgator.com.br/blog/ruby-linguagem-de-programacao/>. Acesso em: 9 mai. 2022.

CLARK, J. Top 10 frameworks para desenvolver seu backend. **Blocg Back4App,** 31 de outubro de 2020. Disponível em:< https://blog.back4app.com/pt/melhores-frameworks-para-desenvolver-backend/>. Acesso em: 9 mai. 2022.

Conheça o Android, o sistema operacional móvel do Google. (s.d.). **TechTudo**, 2022. Disponível em: <https://www.techtudo.com.br/tudo-sobre/android/>. Acesso em: 14 mai. 2022.

Desenvolvimento web: entenda todo o processo. **UseMobile**, 3 de fevereiro de 2022. Disponível em: <https://usemobile.com.br/desenvolvimento-web-processo/>. Acesso em: 9 mai. 2022.

DESKTOP. In: Oxford Languages, 2022. Google, 27 de outubro de 2022. Disponível em: <https://www.google.com/search?q=o+que+é+desktop>. Acesso em: 2 jun. 2022.

Diagrama de caso de uso UML: O que é, como fazer e exemplos. **Lucidchart**, 17 de junho de 2020. Disponível em: <https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>. Acesso: 9 abril. 2022.

ENE, A. C. What is Express.js? Stack **Overdlow**, 15 de fevereiro de 2017. Disponível em: <https://stackoverflow.co

m/questions/12616153/what-is-express-js>. Acesso em: 9 mai. 2022.

ESTRELLA, C. F. O que é JavaScript. **Hostinger**, 19 de agosto de 2021. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-javascript>. Acesso em: 9 mai. 2022.

ESTRELLA, C. F. O Que é PHP? Guia Básico de Programação PHP. **Hostinger**, 22 de janeiro de 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-php-guia-basico>. Acesso em: 9 mai. 2022.

ETTINGER, A. What is Express.js. **Stack Overflow**, 27 de setembro de 2021. Disponível em: <https://stackoverflow.com/questions/12616153/what-is-express-js#:~:text=This%20is%20over,3x/api.html>. Acesso em: 9 mai. 2022.

FERREIRA, C. Laravel Blade. **Blog Especializa TI**, 16 de outubro de 2018. Disponível em: <https://blog.especializati.com.br/laravel-blade/>. Acesso em: 9 mai. 2022.

GONÇALVES, A. O que é CSS? Guia Básico para Iniciantes. **Hostinger**, 14 de fevereiro de 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>. Acesso em: 9 mai. 2022.

HOLOWAYCHUK, T., CASWELL, T., & WILSON, D. Connect: Connect is a middleware layer for Node.js. **Github**, 2022. Disponível em: <https://github.com/senchalabs/connect>. Acesso em: 9 mai. 2022.

Java: o que é, para que serve e por que preciso dele. **Mentorama**, 23 de março de 2021. Disponível em:<https://mentorama.com.br/blog/java-o-que-e-para-que-serve-e-porque-preciso-dele/>. Acesso em: 9 mai. 2022.

KADAMBARI, G., HAN, K. Z., TILAK, S. G., & HUONG, V. V. A Fresh Graduate’s Guide to Software Development Tools and Technologies. **Computing - National University of Singapore**, abril de 2012. Disponível em: <https://www.comp.nus.edu.sg/~seer/book/2e/Ch09.%20Mobile%20Platform.pdf>. Acesso em: 9 mai. 2022.

LI, D. HarmonyOS (HongMeng OS): Everything you need to know. **Huawei Central**, 4 de outubro de 2021.Disponível em: <https://www.huaweicentral.com/harmonyos-hongmeng-os-everything-you-need-to-know/>. Acesso em: 16 jun. 2022.

Linguagem Swift: entenda ela e sua importância na programação. **DigitalHouse**, 14 de junho de 2021. Disponível em: <https://www.digitalhouse.com/br/blog/linguagem-swift/>. Acesso em: 9 mai. 2022.

LONGEN, A. S. O Que é HTML? Guia Básico Para Iniciantes. **Hostinger**, 25 de janeiro de 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>. Acesso em: 9 mai. 2022.

MAGNO, R. Padrões IOS. **Nostrum**, 13 de fevereiro de 2017. Disponível em: <https://www.nostrum.com.br/MMXVI/2017/02/13/padroes-ios>/. Acesso em: 9 mai. 2022.

MARDAN, A. What is Express.js? **Stack OverFlow**, 7 de junho de 2013. Disponível em: <https://stackoverflow.com/questions/12616153/what-is-express-js>. Acesso em: 9 mai. 2022.

MARQUES, R. O que é HTML? Entenda de forma descomplicada.**Home Host** , 26 de Julho de 2019. Disponível em: <https://www.homehost.com.br/blog/tutoriais/o-que-e-html/>. Acesso em: 9 mai. 2022.

MCDADE, J. Blade Templates. **Statamic**, 16 de setembro de 2019. Disponível em: <https://statamic.dev/blade>. Acesso em: 9 mai. 2022.

MELO, D. O que é Java? [Guia para iniciantes]. **Tecnoblog**, 2 de março de 2021. Disponível em: <https://tecnoblog.net/responde/o-que-e-java-guia-para-iniciantes/>. Acesso em: 9 mai. 2022.

MELO, D. O que é JavaScript? [Guia para iniciantes]. **Tecnoblog**, 29 de janeiro de 2021. Disponível em: https://tecnoblog.net/responde/o-que-e-javascript-guia-para-iniciantes/>. Acesso em: 9 mai. 2022.

MELO, D. O que é Python? [Guia para iniciantes]. **Tecnoblog**, 26 de janeiro de 2021. Disponível em: <https://tecnoblog.net/responde/o-que-e-python-guia-para-iniciantes/>. Acesso em: 9 mai. 2022.

MOLINA, F. 10 Types of Software Development — Explained. **Baires Dev**, 26 de outubro de 2020. Disponível em: <https://www.bairesdev.com/blog/10-types-of-software-development-explained>/. Acesso em: 9 mai. 2022.

NOLETO, C. Engenheiro(a) de software: o que é, salário e formação! **Blog Be Trybe**, 31 de maio de 2020. Disponível em: <https://blog.betrybe.com/carreira/engenheiro-de-software/>. Acesso em: 9 mai. 2022.

NOLETO, C. Framework: o que é, como ele funciona e para que serve. **Blog Be trybe**, 13 de fevereiro de 2020. Disponível em: <https://blog.betrybe.com/framework-de-programacao/o-que-e-framework/>. Acesso em: 9 mai. 2022.

NOLETO, C. Paradigmas de programação: o que são e quais os principais. **Blog Be Trybe**, 26 de junho de 2020. Disponível em: <https://blog.betrybe.com/tecnologia/paradigmas-de-programacao/>. Acesso em: 9 mai. 2022.

NOLETO, C. POO: tudo sobre Programação Orientada a Objetos. **Blog Be Trybe**, 9 de agosto de 2020. Disponível em: <https://blog.betrybe.com/tecnologia/poo-programacao-orientada-a-objetos/>. Acesso em: 9 mai. 2022.

NOLETO, C. Ruby: tudo sobre essa linguagem orientada a objetos. **Blog Be Trybe**, 18 de abril de 2020. Disponível em: <https://blog.betrybe.com/linguagem-de-programacao/ruby/>. Acesso em: 9 mai. 2022.

NOLETO, C. UML: o que é, para que serve e quando usar essa linguagem de notação. **Blog Be Trybe**, 27 de julho de 2020. Disponível em: <https://blog.betrybe.com/tecnologia/uml/>. Acesso em: 9 mai. 2022.

ROVEDA, U. Desenvolvimento Web: o que é e como ser um desenvolvedor web. **Kenzie**, 11 de dezembro de 2020. Disponível em: https://kenzie.com.br/blog/desenvolvimento-web/>. Acesso em: 9 mai. 2022.

ROVEDA, U. JavaScript: o que é, para que serve e como funciona o JS. **Kenzie**, 28 de outubro de 2020. Disponível em: <https://kenzie.com.br/blog/javascript/>. Acesso em: 9 mai. 2022.

ROVEDA, U. O que é Python, para que serve e por que aprender. **Kenzie**, 13 de outubro de 2020. Disponível em: <https://kenzie.com.br/blog/o-que-e-python/>. Acesso em: 9 mai. 2022.

ROVEDA, U. O que é back end, para que serve e como aprender em 2021. **Kenzie**, 15 de janeiro de 2021. Disponível em:<https://kenzie.com.br/blog/back-end/>. Acesso em: 9 mai. 2022

ROVEDA, U. As 11 linguagens back-end mais usadas para programar. **Kenzie**, 4 de fevereiro de 2022. Disponível em: <https://kenzie.com.br/blog/linguagens-de-programacao-backend/>. Acesso em: 9 mai. 2022.

SHAHZAD, M.2 de abril de 2016. What is Express.js. **Stack Overflow**, 2 de abril de 2016. Disponível em: <https://stackoverflow.com/questions/12616153/what-is-express-js >. Acesso em: 9 mai. 2022.

SOUTO, M. O que é front-end e back-end. **Alura**, 25 de setembro de 2019. Disponível em: <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>. Acesso em: 9 mai. 2022.

SUGDEN, S. A short guide to Connect Middleware. **Stephen Sugden's website,** 12 de novembro de 2011. Disponível em: <https://stephensugden.com/middleware\_guide/>. Acesso em: 9 mai. 2022.

THIAGO. Conhecendo a Linguagem Ruby. **DevMedia**, 2008. Disponível em: <https://www.devmedia.com.br/conhecendo-a-linguagem-ruby/8226>. Acesso em: 9 mai. 2022.

ZIVADINOVIC, M. What is Express.js. **Stack Overflow**, 30 de abril de 2015. Disponível em: <https://stackoverflow.com/questions/12616153/what-is-express-js>. Acesso em: 9 mai. 2022.