Introdução

Desde a revolução tecnológica iniciada no século XX a internet modificou a perspectiva do mundo juntamente com o andamento de tecnologias que avançassem a demanda de trabalhos por meios tecnológicos pelos negócios. A demanda do mercado por tecnologias melhoradas ao negócio se tornou nítida e essenciais, contudo, o desuso dessas mesmas ocasionará o retrocesso de negócios, juntamente com o atraso de investimentos, aumentando justificavelmente os gastos por tais.

Ao mencionar a modernização e fluidez dos lucros nos meios dos negócios não se limita ao uso dos computadores para apenas o acesso ao mundo virtual, sendo que tal ato de acessar informações nas redes facilitou diversas áreas da vida dos cidadãos, assim como a área 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 acabou por ser extremamente necessário, mesmo após o retorno parcial das atividades, trabalhos presenciais acabaram por serem substituídos por aplicativos (apps) 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 evoluiria. Entretanto, com novos dados levantados, houve um novo consenso de que a tecnologia como um aplicativo de treino e equipamentos de monitoramento ajudou a criar paradigmas neste ramo. Os aparelhos celulares se tornaram uma parte fundamental nos treinos das pessoas, seja eles solitários ou em academias equipadas com aparelhos, criando oportunidades.

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 o uso 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 (aplicativo móvel) para auxiliar os clientes durante o tempo de suas atividades físicas, como método de resolução para que cada cliente possa realizar a prática de alguma atividade física diante do retorno parcial, em conjunto de evitar a contaminação, dispondo que cada individuo poderá consultar seus próprios treinos e atividades por meio de seu dispositivo móvel, deixando mais eficiente e prático o tempo disposto para a realização, proporcionando em conjunto a diminuição dos custos do profissional aplicando métodos mais evolutivos de comunicação optando pelo uso da tecnologia para auxiliar em meios de lucros para seu negócio, em especifico, os profissionais da área de educação física.

Acredita-se com criação de uma aplicação que auxiliará duas partes importantes do processo de melhoria do condicionamento físico do cliente, onde será:

* Para o cliente:

Visibilidade das atividades físicas oferecidas para o seu condicionamento físico;

Maior comunicação com o profissional do campo (Professor da Academia disposta) auxiliando qualquer tipo de dúvida sobre os métodos aplicados;

Independência e aprendizado para a realização de atividades físicas;

Consistência para a evolução significativa de seu condicionamento;

Menos riscos de contaminação enquanto realiza a prática de atividades.

* Para o profissional:

Organização dos treinos passados a cada cliente;

Redução dos custos para o acesso de informações;

Facilidade de comunicação entre o cliente.

Revisão Bibliográfica

Linguagens de programação

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, incluindo dizer que para a construção de uma página estática, construída com HTML e CSS, realizar ações especificas determinadas de acordo com o *script*. As linguagens podem rodar em dois ambientes divergentes, como:

* Back-end;
* Front-end.

Podemos constar exemplos de Linguagens que funcionam nos ambientes acima, assim:

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

Logotipo

Descrição gerada automaticamentePHP

De acordo com [Hostinger](https://www.hostinger.com.br/tutoriais/o-que-e-php-guia-basico), 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.

Umas de suas características predominantes citadas em [DevMedia](https://www.devmedia.com.br/php/) que mais ganham destaque 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.

Um exemplo de código em PHP:

Texto

Descrição gerada automaticamente

JavaScript

JS é a sigla dada a JavaScript, sendo mencionado por [Kenzie](https://kenzie.com.br/blog/javascript/) e [Hostinger](https://www.hostinger.com.br/tutoriais/o-que-e-javascript) que JavaScript é 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.

Dito pelo [Tecnoblog](https://tecnoblog.net/responde/o-que-e-javascript-guia-para-iniciantes/), 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;
* Jogos eletrônicos.

Um exemplo de código em JavaScript:

Texto

Descrição gerada automaticamente

Java

Java é uma linguagem de programação de extrema popularidade onde, a possibilidade de escrever o código apenas uma única vez e rodá-lo em diferentes dispositivos passou a ser implementada em diferentes ambientes de acesso como sites, computadores, *datacenters*, celulares e até mesmo calculadoras.

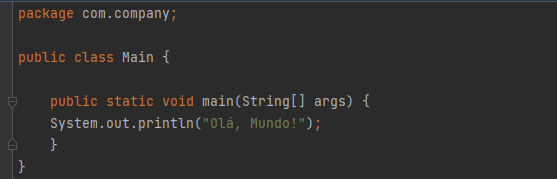
Sendo uma linguagem baseada em Classes e Orientação a objetos possuindo foco em segurança, portabilidade e alta performance, outras de suas características que se destacam é a similaridade de sintaxe a C/C++ com biblioteca de rotinas e APIs para trabalhar com recursos de rede juntamente com gerenciamento 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 elas:

* 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.

*Baseado em* [*Tecnoblog*](https://tecnoblog.net/responde/o-que-e-java-guia-para-iniciantes/) *e* [*Mentorama*](https://mentorama.com.br/blog/java-o-que-e-para-que-serve-e-porque-preciso-dele/)

Um exemplo de código em Java:



Ruby

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.

Como foi citado em [Hostgator](https://www.hostgator.com.br/blog/ruby-linguagem-de-programacao/), uma de suas características de grande destaque é que Ruby é uma linguagem de simples compreendimento, possuindo uma semântica simples além de garantir um grande bônus como o aumento de produtividade aos programadores que a utilizam.

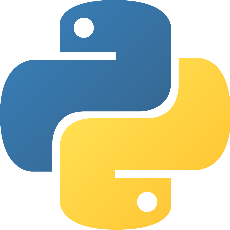
De acordo com [Devmedia](https://www.devmedia.com.br/conhecendo-a-linguagem-ruby/8226) Ruby é uma linguagem ainda nova no mercado, sendo criada no ano de 1995 por *Yukihiro Matsumoto*, este que se baseou em criar uma linguagem de *script* mais poderosa do que Perl e mais orientada a objetos do que Python assim o profissional criou a linguagem Ruby, a atualização mais recente está em sua versão 3.0.0.

Utilizada principalmente no 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.

*Demais fontes utilizadas dos sites* [*Betrybe*](https://blog.betrybe.com/linguagem-de-programacao/ruby/)*,* [*TreinaWeb*](https://www.treinaweb.com.br/blog/o-que-e-a-linguagem-de-programacao-ruby) *e* [*Mentorama*](https://mentorama.com.br/blog/java-o-que-e-para-que-serve-e-porque-preciso-dele/)

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamenteUm exemplo de código em Ruby:

Python

De acordo com [Kenzie](https://kenzie.com.br/blog/o-que-e-python/), Python é uma linguagem de programação de alto nível, denominada também como *High Level Language*, sendo uma das linguagens mais próximas da língua humana 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, como:

* Dinâmica;
* Interpretada;
* Modular;
* Multiplataforma;
* Orientada a Objetos.

Possuindo 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 em [Tecnoblog](https://tecnoblog.net/responde/o-que-e-python-guia-para-iniciantes/), 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.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamenteUm exemplo de código em Python:

Swift

Referência por [Zup](https://www.zup.com.br/blog/swift-linguagem-de-programacao-ios), 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.

Sendo 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.

Citando 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;
* Iteradores e Clausuras.

*Demais fontes utilizadas dos sites* [*Wikipedia*](https://pt.wikipedia.org/wiki/Swift_(linguagem_de_programação)) *e* [*DigitalHouse*](https://www.digitalhouse.com/br/blog/linguagem-swift/)

Um exemplo de código em Swift:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Plataformas de Desenvolvimento

Web

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. *Baseado em* [*Kenzie*](https://kenzie.com.br/blog/desenvolvimento-web/) *e* [*UseMobile*](https://usemobile.com.br/desenvolvimento-web-processo/)

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Front-end

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:

* HTML;
* CSS;
* JavaScript.

Além de possuir consigo a utilização de bibliotecas entre os profissionais do campo, sendo elas:

* React;
* JQuery;
* Node.js.

*Interface gráfica do usuário

Descrição gerada automaticamenteBaseado em* [*Kenzie*](https://kenzie.com.br/blog/desenvolvimento-web/)

HTML

Segundo [Wikipedia](https://pt.wikipedia.org/wiki/HTML) a sigla HTML é dada a *HyperText Markup Language*, ou seja, o HTML é uma Linguagem de Marcação de Hipertexto. Sendo reconhecido como um componente básico da programação Web,segundo o [HomeHost](https://www.homehost.com.br/blog/tutoriais/o-que-e-html/) permitindo inserir o conteúdo e estabelecer a estrutura básica de um *Website*, permanecendo uma organização no conteúdo estabelecido pelo programador.

O [HomeHost](https://www.homehost.com.br/blog/tutoriais/o-que-e-html/) afirma que “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”.

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.

Em 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. Baseado em [HomeHost](https://www.homehost.com.br/blog/tutoriais/o-que-e-html/) e [Hostinger](https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos)

Texto

Descrição gerada automaticamenteUm exemplo de código de HTML:

*Código Fonte retirado em* [*Hostinger*](https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos) *e estilizado no VSCode*

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.

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 eles:

* Estilo Interno: Nesse estilo o 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;
* Estilo Externo: 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;
* Estilo Inline: Nesse estilo é 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.

*Baseado em* [*Hostinger*](https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css)

Texto

Descrição gerada automaticamenteUm exemplo de código CSS:

HTML ✖ CSS ✖ JavaScript

Em conjunto essas linguagens fazem a trindade principal de linguagens utilizadas para o Desenvolvimento Web, onde cada uma abriga sua especularidade de forma que atribua todos os códigos necessários para a 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, como por exemplo:

* HTML: Linguagem de Marcação responsável por estruturar os elementos da página, de forma simplificada e organizada para visualização;
* CSS: 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;
* JavaScript: 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.

Inserindo com o exemplo dado do corpo humano, podemos obter a seguinte exemplificação da trindade de linguagens:

*Baseado em* [*Alura*](https://www.alura.com.br/artigos/html-css-e-js-definicoes)

Back-end

O Back-end é a área da programação voltada para parte lógica de sites, aplicativos, softwares ou outros tipos de sistemas de informação, garantindo seu bom funcionamento e seus objetivos de acordo com a necessidade.

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.

Sendo as principais linguagens de programação, podemos citar algumas como:

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

Além utilizar alguns frameworks, como:

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

*Baseado em* [*Kenzie*](https://kenzie.com.br/blog/back-end/)*(1),* [*Kenzie*](https://kenzie.com.br/blog/linguagens-de-programacao-backend/)*(2),* [*BlogBack*](https://blog.back4app.com/pt/melhores-frameworks-para-desenvolver-backend/)

Desktop

A Desktop (no português, área de trabalho) 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. Ela proporciona esses benefícios à rotina do indivíduo, dispondo uma ótima estrutura para se concentrar nas atividades que deseja desenvolver sem nenhum impedimento.

*Baseado em Oxford Languages*

Imagem digital fictícia de personagem de jogo de vídeo game

Descrição gerada automaticamente com confiança baixaImagem digital fictícia de personagem de desenho animado

Descrição gerada automaticamente com confiança médiaInterface gráfica do usuário, Site

Descrição gerada automaticamente

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.

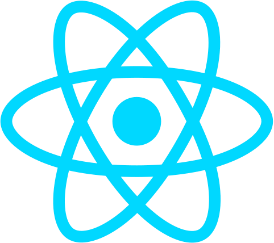
Os frameworks é criado por comunidades de desenvolvedores que orientam a consistência e rigorosidade, modificando e revisando o código para um padrão de qualidade, um dos exemplos dessa qualidade seria a legibilidade da construção do código, um código limpo é importante para os desenvolvedores para que os desenvolvedores realizem as manutenções e entender do que se trata ao ler o código.

Sua vantagem 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.

Alguns Frameworks mais utilizados para o desenvolvimento são:

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

*Baseado em*[*Back4app*](https://blog.back4app.com/pt/melhores-frameworks-para-desenvolver-backend/) *e* [*Betrybe*](https://blog.betrybe.com/framework-de-programacao/o-que-e-framework/)

React Native

Introduzida pelo Facebook em 2018, é uma framework JavaScript para criar aplicativos nativos para qualquer plataforma e dispositivo. Com ela, pode se desenvolver aplicativos para ambas as plataformas Android e iOS. Ela permite também criações de versões específicas da plataforma de vários componentes, permitindo o fácil uso da base de código em plataformas múltiplas.

Alguns de seus recursos são:

* desenvolvimento *low-code* (criação de software por meio de UI);
* há compatibilidade com plugins de terceiros;
* API declarativa para IU preditiva.

*Baseado em Clariontech*

Xamarin

Introduzida 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.

Alguns de seus recursos são:

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

*Baseado em* [*Clariontech*](https://www.clariontech.com/blog/top-mobile-app-development-frameworks-in-2019)

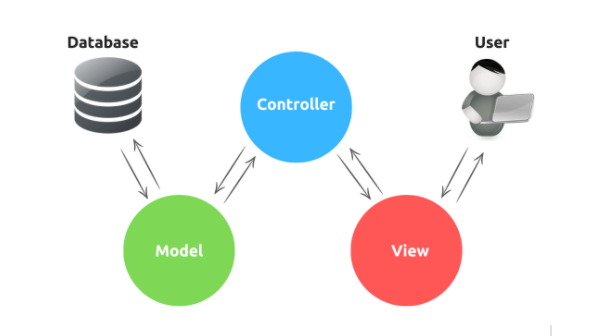
Swiftic

Introduzida 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.

*Baseado em* [*Clariontech*](https://www.clariontech.com/blog/top-mobile-app-development-frameworks-in-2019)

Ruby on Rails

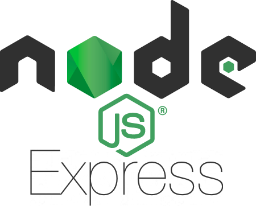
De acordo com [TreinaWeb](https://www.treinaweb.com.br/blog/o-que-e-ruby-on-rails), Ruby on Rails É um framework de código aberto escrito em Ruby, seguindo o padrão arquitetural MVC (Model-View-Controller) que, recebendo uma requisição, solicita ao Model informações necessárias, que quando retornam ao Controller, as envia para a View que as renderizará.



Fonte da imagem: [cloudfront.net](https://dkrn4sk0rn31v.cloudfront.net/uploads/2020/08/MVC.png)

* O *Model* é 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.
* A *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.
* 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.

A vantagem de usá-lo é por ser 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.

Express.js

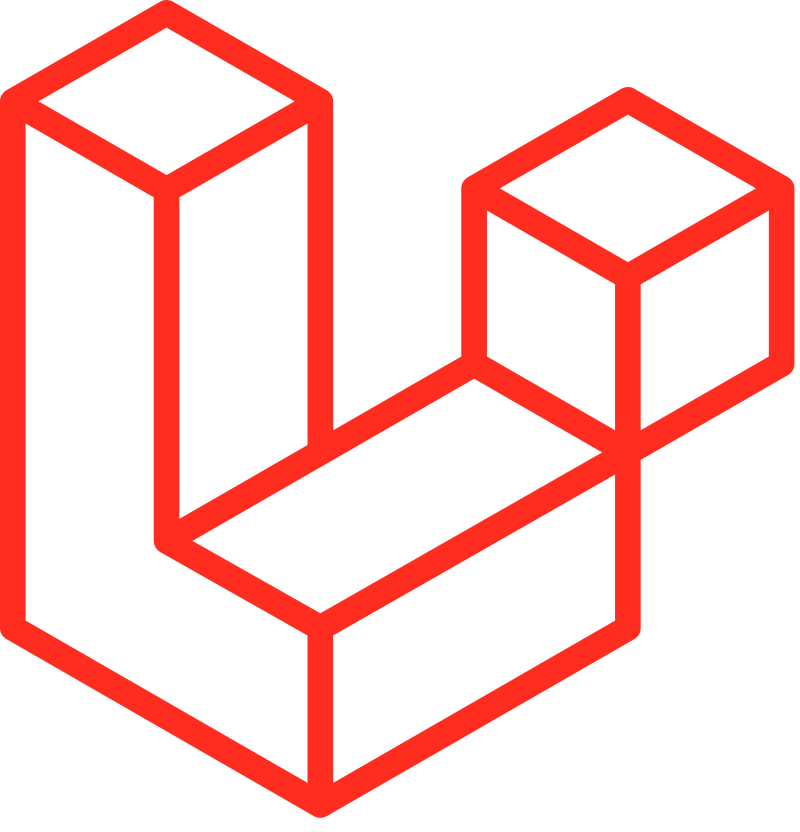
É um framework para Node.js. Ele é projetado a construir página única, página múltipla e aplicações de web híbridas, como um esqueleto, se usa aplicativos baseados em servidor(es) que recebe solicitações de entrada dos usuários. Ele é construído a partir da configuração e simplicidade do middleware Connect,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

Connect é 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.

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 regulares que terá que ser reimplementado. No caso, usando o Express.js, ele só vai estar lá para ser usado.

Redis é um armazenador de chave-valor 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.

*Baseado nas respostas de chovy, mzdv e Andrei-Cristian Ene em* [*stackoverflow*](https://stackoverflow.com/questions/12616153/what-is-express-js) *, no github de* [*senchalabs*](https://github.com/senchalabs/connect) *e no site* [*stephensugden*](https://stephensugden.com/middleware_guide/)

Laravel

É um framework que é desenvolvido em PHP desenvolvido por Taylor Otwell, pela 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.

Seu objetivo seria resolver os problemas que têm atuações genéricas, isso permite o 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 eficas. 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 os arquivos para reaproveitar 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, podendo usar configuradores de parâmetros explícitos, passando dados contextuais, buscar a saída de um loop e paginação.
  + As seções injetam o layout de conteúdo pai de um layout extendido do blade filho.
* O módulo de autenticação ou autorização completo é o que provê todos os recursos para a sua implementação, como o cadastro, a autenticação e autorização de usuários, recuperação de senhas, logout e sessões.

Com o Laravel, não se 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 intruções SQL. Ele implementa o padrão Active Record, onde o model da aplicação representa a tabela de um BD.

*Baseado em* [*TreinaWeb*](https://www.treinaweb.com.br/blog/o-que-e-laravel)*,* [*Especializati*](https://blog.especializati.com.br/laravel-blade/) *e* [*Statamic*](https://statamic.dev/blade#system-variables)

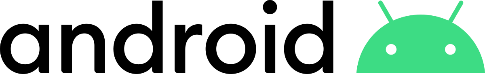
Plataformas Mobile

O 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.

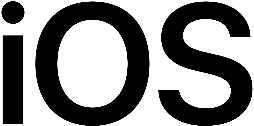
As plataformas mobile (ou também sistemas operacionais mobile) são aqueles que permitem o usuário rodar softwares e serviços desejáveis no seu dispositivo móvel, também fornecem ferramentas que permite aplicativos compartilharem dados e serviços,

Algumas plataformas mais utilizadas são:

* iPhone;
* Android.

Android

É uma plataforma da Google inicialmente desenvolvida pela Android Inc. 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.

 iOS

É uma plataforma da Apple desenvolvida nos EUA presente apenas nos dispositivos da Apple, ela se consiste em quatro camadas, delas:

* Core OS;
* Core Services;
* Media;
* 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.
* Na camada Core Services, contém os serviços fundamentais do sistema, onde, mesmo que desenvolvedor não utilizar os serviços diretamente, o sistema construirá em cima, 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.
* 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.
* XML Support oferece suporte adicional para a manipulação .XML, podendo também transformá-lo para HTML.
* Na camada Media, contém o conteúdo visual e sonora, como vídeos, fotos e áudios, ela foi projetada para impulsionar a implementação dos aplicativos multimídia.
* Na camada Cocoa Touch, contém os frameworks principais para a construção de aplicativos dessa camada, ela define a infra-estrutura para tecnologias fundamentais, como multitarefa, o serviço de notificação e os serviços de alto nível do sistema.

Paradigmas de Programação

Segundo o site: betrybe, 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: betrybe.

Os Paradigmas são classificados, de acordo com a betrybe.com, em alguns tipos como por exemplo:

* Imperativo;
* Declarativo;
* Funcional;
* Lógico;
* Orientado a Objetos;
* Desenho de uma pessoa

  Descrição gerada automaticamente com confiança médiaOrientado a Eventos.

Paradigma Imperativo

Como o blog da betrybe menciona, 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.

Algumas linguagens que abordam este paradigma são:

* C;
* C++;
* Java;
* Pascal.

***O foco da resolução estará em “Como” deve ser feita.***

***Imagem digital fictícia de personagem de desenho animado

Descrição gerada automaticamente com confiança média***

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.

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, de acordo com a betrybe.com.

Algumas linguagens que abordam este paradigma são:

* Absys;
* Ciao;
* Alice.

***O foco da resolução estará em “O quê” deve ser solucionado.***

Desenho de personagem de desenho animado

Descrição gerada automaticamente com confiança baixa

Paradigma Funcional

Segundo a fonte betrybe, podemos inferir que 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.

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;
* JavaScript.

Uma imagem contendo Calendário

Descrição gerada automaticamente

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.

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, de acordo com betrybe.com:

* QLISP;
* Mercury;
* Prolog.

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Paradigma Orientado a Objetos

O Paradigma Orientado a Objetos ou POO, de acordo com betrybe.com, 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.

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, como:

* Herança;
* Polimorfismo;
* Encapsulamento;
* Abstração.

Algumas linguagens que abordam este paradigma são:

* Java;
* C++;
* C#;
* Python.

Interface gráfica do usuário

Descrição gerada automaticamente

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, concluído através do blog betrybe.com.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Engenharia de Software

Inferindo-se através do site: betrybe.com, 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, de acordo com a betrybe.

As engenharias de software são divididas em algumas, sendo as principais:

* UML;
* Diagrama de Classes;
* Casos de Uso.

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

UML

De acordo com a informação adquirida no blog betrybe, 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 geométrica de retângulos, linhas, setas e balões – dos quais empenham o papel de espelhar componentes de um aplicativo, suas interações e mudanças.

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, concluindo-se através da fonte: betrybe.com.

Há profusos tipos de diagramas UML e eles se diferem em duas castas, segundo o site betrybe: 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.

Diagrama de Classes

Os Diagramas de Classes são divididos em categorias, sendo eles:

* Diagramas Estruturais;
* Diagramas Comportamentais.

Diagramas Estruturais

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 advêm 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.

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.

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.

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 usuária com cada uma. Se trata de um diagrama simples e não requer grande conhecimento para operá-lo.

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.

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.