SENAI

Evellyn Achnitz de Oliveira

Murilo Henrique Ribeiro da Silva

Pedro Francisco da Fonseca Oliveira Alves

Tuanny Cristina Thomazelli

**FRAMEWORK**

Node.js com Express.js

Sorocaba

2025

Evellyn Achnitz de Oliveira

Murilo Henrique Ribeiro da Silva

Pedro Francisco da Fonseca Oliveira Alves

Tuanny Cristina Thomazelli

**FRAMEWORK:**

Node.js com Express.js

Trabalho da matéria PRJ (Projetos) no qual o objetivo e apresentar sobre frameworks especificadamente o Note.js com Express.js

Vedilson Prado

Sorocaba-SP

2025

**SUMÁRIO**

[INTRODUÇÃO SOBRE A IMPORTÂNCIA DOS FRAMEWORKS PARA O MUNDO DO TRABALHO 3](#_Toc189685990)

[EXPLICAÇÃO SOBRE O QUE É UM FRAMEWORK E SUAS PRINCIPAIS FUNÇÕES 4](#_Toc189685991)

[APRESENTAÇÃO SOBRE OS DIFERENTES TIPOS DE FRAMEWORK 4](#_Toc189685992)

[Diferentes arquiteturas 5](#_Toc189685993)

[DESCRIÇÃO SOBRE O FRAMEWORK - Node.js com Express.js 5](#_Toc189685994)

[Express.js 6](#_Toc189685995)

[Instalação 6](#_Toc189685996)

[APRESENTAÇÃO DE EXEMPLOS DE CASOS DE USO DO FRAMEWORK DEFINIDO 8](#_Toc189685997)

[Vantagens 9](#_Toc189685998)

[Desvantagens 9](#_Toc189685999)

[CONCLUSÃO 9](#_Toc189686000)

[REFERÊNCIAS 9](#_Toc189686001)

1. INTRODUÇÃO SOBRE A IMPORTÂNCIA DOS FRAMEWORKS PARA O MUNDO DO TRABALHO

Framework é um conjunto de componentes que se pode  
reutilizar e ajudam a desenvolver o software de forma mais simples e eficaz.  
Ele define a estrutura do projeto e fornece as ferramentas que podem auxiliar esse processo para criar o que o desenvolvedor deseja fazer. Podemos incluir:  
bibliotecas e módulas para tarefas especificas, cuidam de funções como manipulação de dados, gestão de sessões, segurança, autenticação, etc. Dessa forma, através das múltiplas linguagens de programação os Frameworks são criados e identificados pelas metodologias, propósitos e implementações que na maioria das vezes poderão reutilizar suas estruturas e códigos.

1. EXPLICAÇÃO SOBRE O QUE É UM FRAMEWORK E SUAS PRINCIPAIS FUNÇÕES

Dentre as diversas funções dos frameworks podemos citar:  
definir a estrutura do projeto, fornecer ferramentas para o desenvolvimento e  
simplificar seu processo, aumentar a produtividade, promover a criação de  
aplicativos mais robustos e escaláveis, melhorar a qualidade do código e  
facilitar sua manutenção a longo prazo, garantir a segurança e compatibilidade do código gerado.

1. APRESENTAÇÃO SOBRE OS DIFERENTES TIPOS DE FRAMEWORK

Angular: Um dos frameworks mais famosos do front-end, concorrente direto do React, o Angular é uma opção muito boa para quem procura um padrão muito usado e com uma comunidade enorme. É uma tecnologia que torna o desenvolvimento mais robusto e legível ao trazer inovações importantes, como Data Binding e maior suporte a testes.

ASP.NET core: Outra opção das mais usadas é o ASP.NET Core, padrão da Microsoft. É uma evolução do ASP.NET, open-source, focado em modernizar o desenvolvimento web.

Flask: O Flask é um framework para o Python que atua no back-end das aplicações web. É conhecido como um microframework, por conta de sua simplicidade e velocidade de funcionamento. Ademais, ele é incrivelmente versátil: importante para pequenos projetos e para aplicações mais robustas.

Django: O Django é uma conhecida alternativa para lidar com Python no back-end. Permite gerenciar microsserviços, manipular bancos de dados, autenticação do usuário, feed RSS, entre outros. Para os bancos de dados, em específico, o Django oferece suportes a diversos tipos relacionais, como PostgreSQL, MySQL e SQLite.

Outro ponto de destaque dessa tecnologia é o seu foco em segurança e proteção dos sites. Ele trabalha para auxiliar no combate a falsas requisições, injeção de SQL e outras investidas comuns em páginas web.

Diferentes arquiteturas

MVC (Model-View-Controller): O MVC é uma arquitetura clássica usada para  
separar as responsabilidades em uma aplicação, facilitando a manutenção e  
escalabilidade do código.

MVVM (Model-View-ViewModel): definição: é um padrão arquitetural de design que visa separar a lógica de apresentação e a interface de usuário de forma mais eficiente do que o padrão MVC (Model-View-Controller), especialmente em aplicações que exigem interação rica com o usuário, como apps móveis e aplicações desktop.

1. DESCRIÇÃO SOBRE O FRAMEWORK - Node.js com Express.js

Node.js é uma plataforma de desenvolvimento que permite a execução de código JavaScript no servidor. Por se tratar de uma linguagem popularmente conhecida para a construção de aplicações web mais interativas, o JavaScript possui grande foco no Front-end (client side), ou seja, é comumente utilizada para rodar no “lado cliente” da aplicação. Mas, com a evolução das tecnologias web, tornou-se possível fazer o JavaScript rodar também no Back-end, e é nesse momento que surge o Node.js.

O Node.js é um ambiente de execução assíncrono do código JavaScript do lado servidor (server side), que na prática se reflete na possibilidade de criar aplicações standalone (autossuficientes) em uma máquina servidora, sem a necessidade do navegador, e trabalhar de forma a não bloquear a aplicação no momento de sua execução, colocando os processos mais demorados para um segundo plano.

Quando executamos um comando escrito em JavaScript, o Node.js interpreta esse comando e faz a sua conversão para a linguagem de máquina a ser executada pelo computador. Por esse motivo, o Node.js também pode ser referido como um JavaScript Runtime, ou um programa de execução do JavaScript.

O Node.js trabalha fortemente com o conceito de módulo, que visa organizar o código em partes pequenas e especializadas, além de aplicar o princípio de encapsulamento ao “esconder” o código e expor somente o necessário a outras partes da aplicação.

4.1 Express.js

Existem diversos frameworks populares que utilizam Node.js, entre eles o Express.js, que é amplamente utilizado para construir APIs e aplicações web. Ele fornece uma variedade de recursos, incluindo roteamento (rotas responsáveis por mapear URLs para funções de manipulação de solicitações, permitindo que você defina como o aplicativo responde a solicitações HTTP específicas), middleware (funções que têm acesso ao objeto de solicitação, ao objeto de resposta e à próxima função no ciclo de solicitação-resposta do aplicativo, usadas para executar tarefas comuns, como registro de solicitações, análise de corpo de solicitação e muito mais) e manipulação de solicitações e respostas HTTP. Com o Express, os desenvolvedores podem criar APIs RESTful, sites dinâmicos e muito mais de forma rápida e eficientes.

4.2 Instalação

O Node.js é uma ferramenta de código aberto e gratuita. Ela também é multiplataforma, o que nos permite entregar uma solução para rodar em ambientes Windows, Linux ou MacOS, e seu interpretador é baseado no V8 da Google (um interpretador JavaScript desenvolvido pela Google e utilizado pelo Chrome. Ele também é conhecido como a máquina virtual do JavaScript. Foi desenvolvido usando a linguagem C++, é de código aberto e nasceu com a intenção de acelerar a execução de aplicações desenvolvidas em JavaScript).

Para essa instalação, nós vamos usar a versão LTS (Long-term support, ou suporte de longo prazo, em português) porque é a mais estável. A versão Current é a mais atualizada, porém pode conter alguns bugs.

No processo de instalação do Node.js nos é entregue também o NPM (Node Package Manager), que vai nos permitir compartilhar os módulos que criamos com a comunidade além de possibilitar a reutilização de módulos criados por outras pessoas desenvolvedoras. NPM é uma tecnologia responsável pelo gerenciamento e execução desses pacotes e dependências. Através do NPM podemos gerenciar todas dependências dos projetos JavaScript que estamos desenvolvendo, com ele temos acesso a uma gama enorme de bibliotecas e frameworks JS.

Para a instalação no Windows, vamos começar acessando a página de download do Node.js, ao clicar na opção Windows Installer da versão LTS, será iniciado o download automático do pacote instalador. Quando o download estiver concluído, basta clicar no instalador que foi baixado para que ele seja executado. Uma janela será exibida e iremos prosseguir clicando em Next, chegando à tela de licenças. Aqui basta preencher a checkbox para concordar com os termos de uso e avançar com o clique em Next novamente. Na tela seguinte, vai aparecer a opção de alterar o local em que o Node será instalado. Optamos por manter o caminho padrão, clique em Next. A seguir, serão escolhidas quais as funcionalidades do Node queremos que sejam instaladas. A recomendação é que não altere nada e faça a instalação de todas, mas caso queira alterar essa configuração, é importante garantir que ao menos as opções Node.js runtime, npm package manager e Add to PATH estejam selecionadas. Em seguida, vai aparecer a opção de instalar as ferramentas para módulos nativos. Em algumas situações específicas, alguns pacotes do npm precisam ser compilados em sua máquina para rodarem no Windows sem impedimentos. Para isso, é necessário que essas ferramentas para módulos nativos estejam instaladas, pois elas irão cuidar da compatibilidade com o Windows. Portanto, para evitar problemas futuros, selecione a checkbox para que seja feita a instalação automática dessas ferramentas. Feito isso, basta selecionar a opção install, que vai aparecer na tela seguinte, e aguardar até que a instalação seja finalizada. Nesse momento, pode ser que algumas janelas do terminal se abram, que é justamente a responsável pela instalação das ferramentas para módulos nativos. Basta clicar em qualquer tecla para continuar e esperar até que seja finalizada. Assim que a instalação terminar, é interessante que você reinicie a máquina, para garantir que Node.js foi escrito corretamente no caminho do PATH do Windows. Mas antes de reiniciar tenha certeza de que a instalação terminou. Por fim, para verificar se a instalação ocorreu corretamente, basta abrir o Prompt de Comando e rodar os comandos node --version. Para verificar a versão do gerenciador de pacotes para Node (o NPM) rode os comandos npm –version. Agora basta digitar node, e você já pode executar JavaScript através do terminal. E a instalação do Node.js foi concluída, além da verificação que ele foi instalado corretamente no Windows.

A instalação no macOS é bastante similar com a instalação do Windows, pois também usa um pacote de instalação. Inicialmente, vamos até o site de download do Node. As versões LTS já estão disponíveis por padrão. Então, selecionamos a opção macOS Installer (.pkg). Ao selecionarmos essa opção, o pacote de instalação será baixado. Assim que o download terminar, clicamos no arquivo baixado e executamos. Depois, podemos prosseguir ao clicar em \*Continue\*, sem nos esquecermos de aceitar os termos de uso. Clique em \*Close\* e agora vamos conferir se o Node e o npm foram realmente instalados, comece digitando node --version e npm –version, se o prompt responder com suas respectivas versões, mostra que conseguimos concluir a instalação do Node e do npm.

1. APRESENTAÇÃO DE EXEMPLOS DE CASOS DE USO DO FRAMEWORK DEFINIDO

O Node.js pode ser usado em APIs, aplicações web em tempo real como chats ou ferramentas de múltiplos usuários, jogos multiplayer, aplicações que demandam alta escalabilidade e servidores de streaming de dados. Empresas como LinkedIn, Walmart, PayPal, Uber e Netflix, usam o Node.js em suas aplicações, destacando benefícios como escalabilidade, velocidade de entrega e performance excepcional.

5.1 Vantagens

- Multiplataforma: podemos criar desde soluções web a aplicações desktop;

- Open Source: como já citado, o Node.js tem seu código aberto, o que nos possibilita contribuir com o projeto e até mesmo realizar customizações;

- Multi-paradigma: com o JavaScript podemos adotar várias formas de codificar usando paradigmas como a orientação a objetos, o funcional e o orientado a eventos.

- Facilitam o desenvolvimento rápido de aplicações backend em Node.js.

- Oferecem recursos pré-construídos, como roteamento, manipulação de solicitações HTTP e autenticação.

- Possuem comunidades ativas de desenvolvedores, o que pode facilitar o suporte e a resolução de problemas.

- Permitem a escalabilidade e o gerenciamento eficiente de recursos.

5.2 Desvantagens

- Alguns frameworks podem ter uma curva de aprendizado inicial mais íngreme, especialmente para desenvolvedores iniciantes em Node.js.

- Alguns frameworks podem ser menos flexíveis em termos de personalização e extensibilidade.

- Dependendo do tamanho do projeto, alguns frameworks podem adicionar uma camada adicional de complexidade e sobrecarga.

1. CONCLUSÃO

O framework Express.js, amplamente utilizado no ecossistema Node.js, destaca-se pela sua simplicidade, flexibilidade e alto desempenho no desenvolvimento de aplicações web e APIs. Através de sua arquitetura minimalista, o Express.js permite aos desenvolvedores criar e gerenciar servidores de forma eficiente, facilitando a implementação de rotas, middleware e outras funcionalidades essenciais para a construção de aplicações modernas. Além disso, sua extensibilidade através de módulos e a vasta comunidade de desenvolvedores contribuem significativamente para o seu contínuo aprimoramento e adoção no mercado.

Em suma, o Express.js consolida-se como uma ferramenta robusta e versátil, ideal para projetos de diferentes escalas e complexidades, sendo uma escolha confiável para desenvolvedores que buscam uma solução eficiente e escalável no desenvolvimento de aplicações com Node.js.

1. REFERÊNCIAS

LENCINA, W. **O que é um framework e para que serve?** Disponível em: <https://ebaconline.com.br/blog/framework-seo>.

ESCOLA DNC. **Frameworks em Programação: conheça os principais e benefícios - Blog DNC**. Disponível em: <https://www.escoladnc.com.br/blog/frameworks-em-programacao-conheca-os-principais-e-beneficios/>. Acesso em: 30 jan. 2025.

GREMES, M. **Framework: o que é e quais as vantagens de utilizar? - Blog MBA Esalq USP**. Disponível em: <https://blog.mbauspesalq.com/2023/08/02/framework-o-que-e-e-quais-as-vantagens-de-utilizar/>. Acesso em: 30 jan. 2025.

KAUR, P. **10 Most Popular Frameworks For Building RESTful APIs**. Disponível em: <https://www.moesif.com/blog/api-product-management/api-analytics/10-Most-Popular-Frameworks-For-Building-RESTful-APIs/>.

**Entendendo o Pattern Model View ViewModel MVVM**. Disponível em: <<https://www.devmedia.com.br/entendendo-o-pattern-model-view-viewmodel-mvvm/18411>>.

SACRAMENTO, Gabriel. **O que é framework em programação**. 2024. Disponível em: https://blog.somostera.com/desenvolvimento-web/o-que-eframework. Acesso em: 30 jan. 2025.

HABBEMA, Hugo. **Node.js: Módulo Express**. 2024. Disponível em: https://medium.com/@habbema/node-js-m%C3%B3dulo-express-aeaea83c853c. Acesso em: 30 jan. 2025.

‌ALURA. **Como instalar o Node.js no Windows, Linux e macOS**. Disponível em: <https://www.alura.com.br/artigos/como-instalar-node-js-windows-linux-macos>. Acesso em: 4 fev. 2025.

‌DEV, L. **O que é Node.js Web Application Frameworks.** Disponível em: <https://lbodev.com.br/glossario/o-que-e-node-js-web-application-frameworks/>. Acesso em: 4 fev. 2025.

**‌Por que aprender Node.js? 8 Casos reais!** Disponível em: <https://www.luiztools.com.br/post/por-que-aprender-nodejs/>. Acesso em: 4 fev. 2025.

‌