



# Node.js

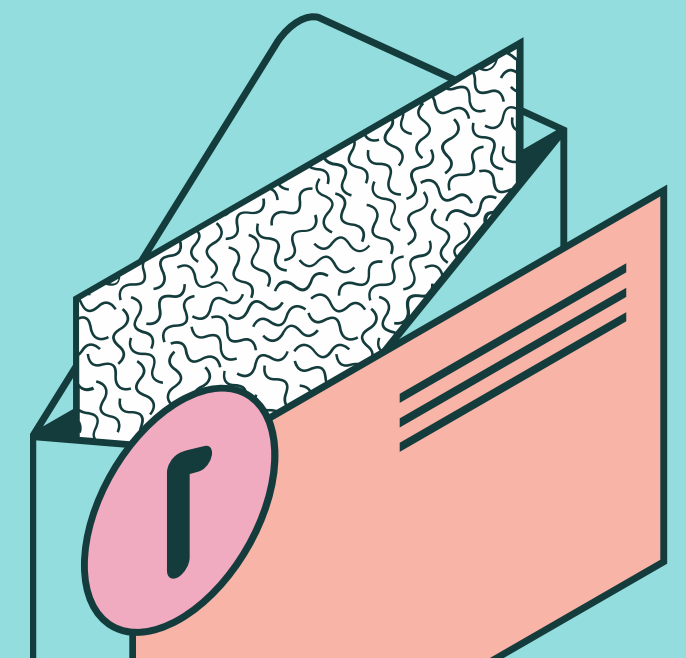
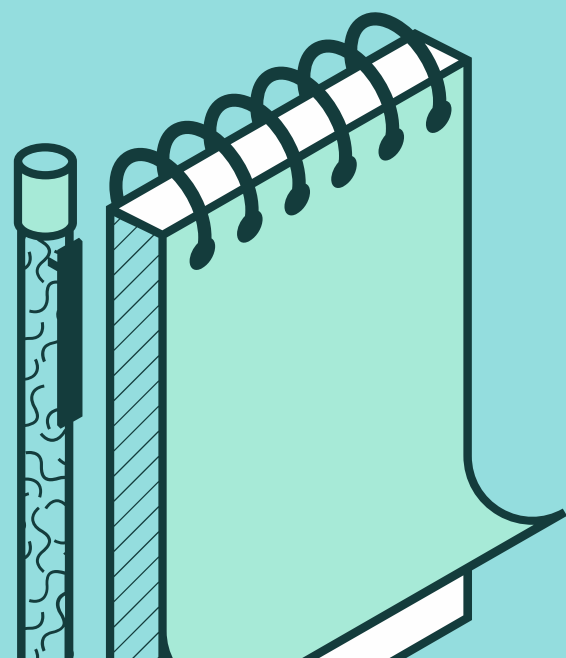
Projeto para o trabalho de programação  
para internet



# Componentes

**Filipe dos Anjos Sousa**

**Thalys Christian de A. Bergmaschi Gonçalves**



# Apresentando o Node.js





# História

- **2009:** Criado por Ryan Dahl, o Node.js nasceu com o intuito de permitir a execução de código JavaScript no servidor, utilizando o motor V8 do Google Chrome.
- **Baseado em JavaScript:** Sua arquitetura assíncrona e orientada a eventos revolucionou o desenvolvimento web, possibilitando o uso do mesmo idioma (JavaScript) tanto no frontend quanto no backend.
- **Ecossistema de Módulos:** O Node.js apresenta um vasto repositório de módulos, impulsionando a construção de aplicativos escaláveis e eficientes.

# História

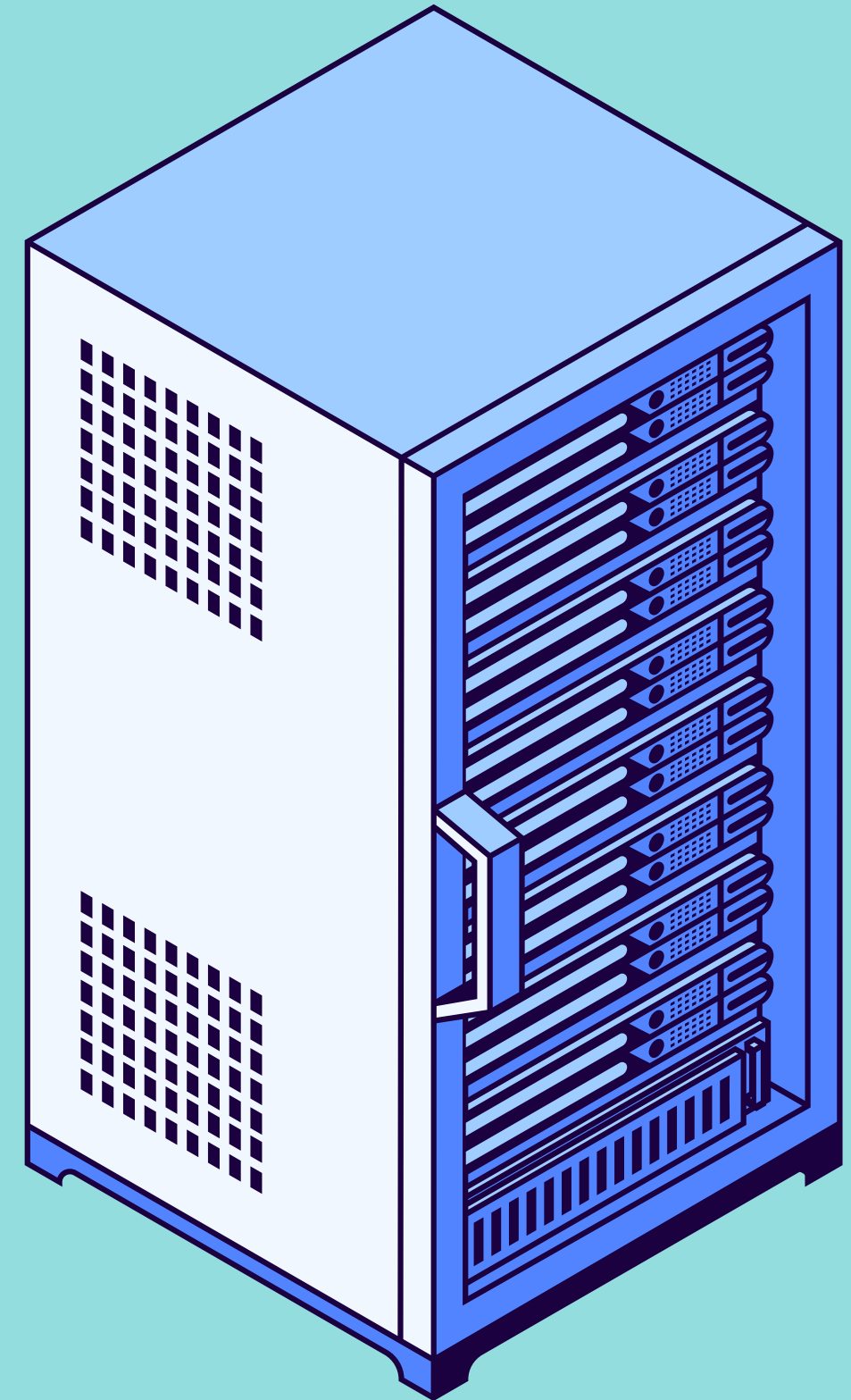


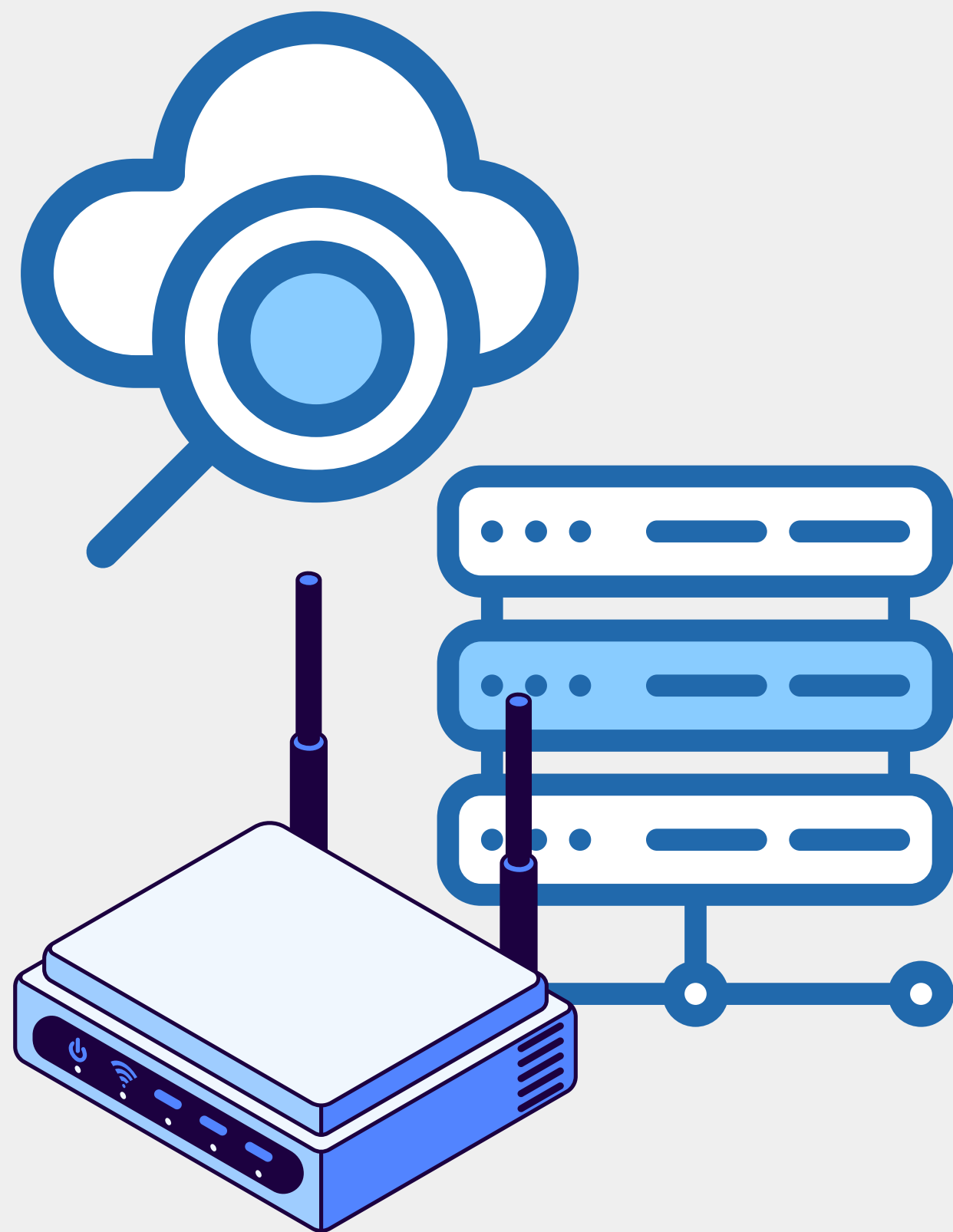
- **Performance e Escalabilidade:** Graças à sua natureza assíncrona, o Node.js se destaca na gestão de muitas conexões simultâneas, proporcionando alta performance e escalabilidade.
- **Adoção pela Comunidade:** Rapidamente ganhou popularidade entre desenvolvedores, empresas e startups, tornando-se uma base fundamental para aplicações web modernas.
- **Evolução Contínua:** Com uma comunidade ativa e atualizações regulares, o Node.js continua a se adaptar às demandas crescentes do desenvolvimento web.



# Principal uso

É amplamente empregado para desenvolver aplicativos web escaláveis e de alto desempenho, particularmente em ambientes que demandam muitas requisições simultâneas. Sua capacidade de lidar com operações de entrada e saída de forma assíncrona e orientada a eventos o torna ideal para construir servidores eficientes, APIs dinâmicas, aplicações em tempo real (como chats e jogos online), microsserviços e diversas soluções que requerem escalabilidade e agilidade.





# NPM (Node Package Manager)

Gerenciador de pacotes extremamente poderoso e central no ecossistema do Node.js. Ele permite que desenvolvedores instalem, gerenciem e compartilhem pacotes de código reutilizáveis. O NPM é fundamental para o desenvolvimento de aplicativos usando Node.js, oferecendo acesso a milhares de bibliotecas e ferramentas prontas para uso. Além disso, ele simplifica a gestão de dependências, automatiza tarefas repetitivas e facilita a distribuição de projetos, contribuindo significativamente para a eficiência e produtividade dos desenvolvedores.

```
{
  "name": "nodejs-project",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "nodemon server"
  },
  "author": "XYZ",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "express": "^4.17.1",
    "mongoose": "^5.6.4",
    "mysql2": "^1.6.5",
    "nodemon": "^1.19.1",
    "pug": "^2.0.4",
    "sequelize": "^5.9.4"
  }
}
```

# Pacotes/Bibliotecas

Os pacotes são unidades de código reutilizáveis distribuídas através do NPM. Esses pacotes abrangem desde pequenas funcionalidades até frameworks completos, disponibilizando uma enorme variedade de bibliotecas prontas para uso. Essenciais para o desenvolvimento, esses pacotes são compartilhados no repositório do NPM e são gerenciados pelo arquivo `package.json`, simplificando a gestão de dependências e permitindo aos desenvolvedores economizar tempo, promover a modularização do código e acelerar o processo de desenvolvimento ao aproveitar funcionalidades já existentes.



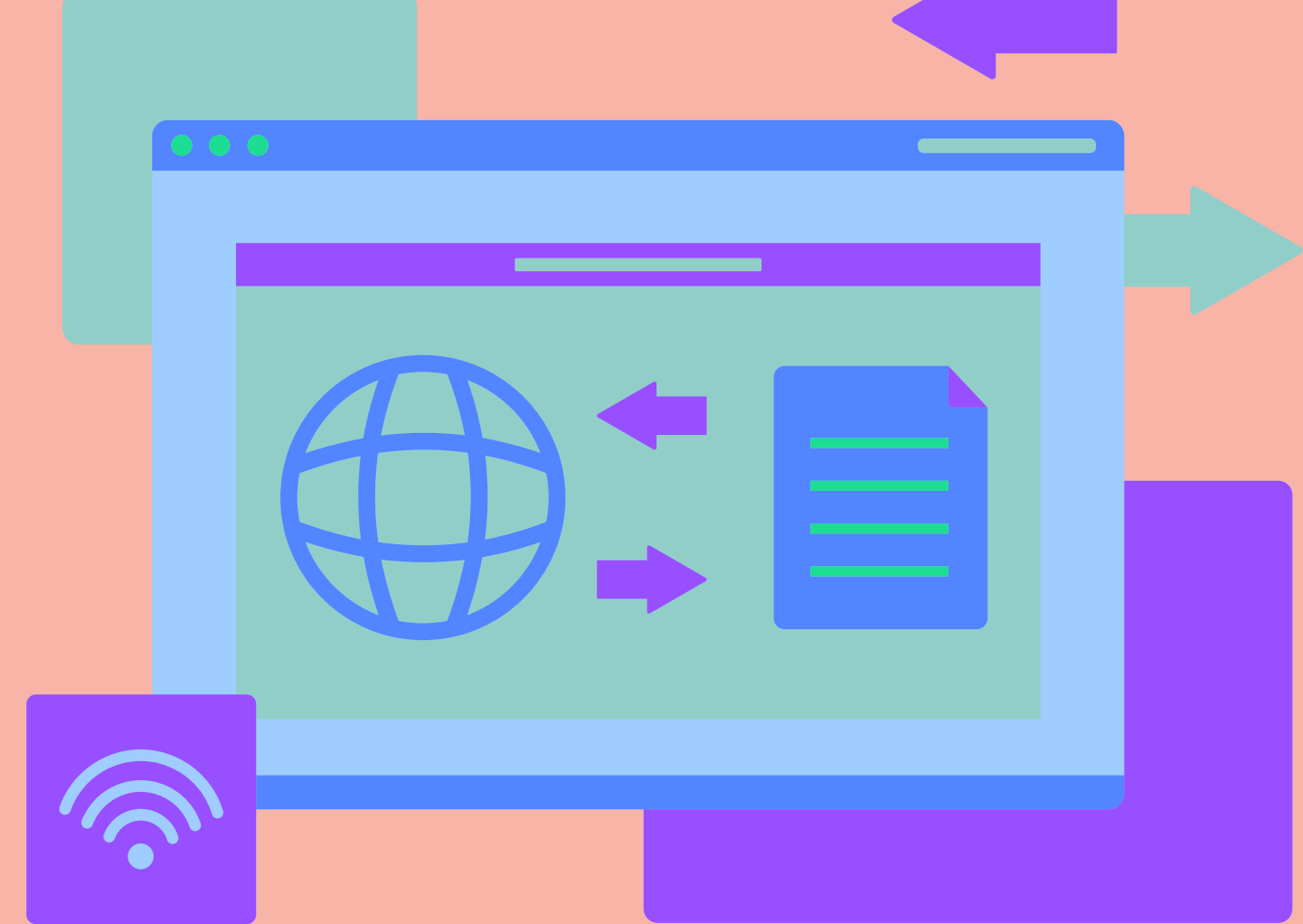
# Express.js

## Framework Minimalista:

O Express.js é um framework web minimalista para Node.js, oferecendo uma estrutura flexível e leve para criar aplicativos web robustos com facilidade.

## Roteamento Simplificado:

Sua funcionalidade de roteamento permite definir de forma clara e organizada como a aplicação responde a diferentes requisições HTTP, facilitando a criação de APIs RESTful e gerenciamento de rotas.



## Ampla Extensibilidade:

Express.js é altamente extensível, permitindo que desenvolvedores incorporem facilmente middlewares de terceiros para adicionar funcionalidades como autenticação, compressão, logging e muito mais.

Express.js, além de sua agilidade e simplicidade, destaca-se pela sua grande comunidade de desenvolvedores ativos e pela vasta gama de recursos adicionais disponíveis. Com uma base de usuários robusta, o framework possui uma biblioteca extensa de middlewares, templates e extensões, oferecendo aos desenvolvedores uma ampla gama de opções para aprimorar e personalizar suas aplicações. Essa comunidade ativa também resulta em constantes atualizações, suporte abrangente e uma rica documentação, facilitando o aprendizado e a resolução de problemas para aqueles que utilizam o Express.js em seus projetos.

Express

JS

# Outros pacotes e frameworks usados

1 ————— 2 ————— 3 ————— 4 ————— 5

## NODEMON

Uma ferramenta utilizada no desenvolvimento Node.js que monitora alterações nos arquivos do projeto e reinicia automaticamente o servidor sempre que há mudanças, agilizando o processo de desenvolvimento..

## REACT-ROUTER-DOM

Um pacote para aplicativos React que facilita a navegação entre diferentes partes da interface, oferecendo um sistema de rotas para construir aplicações de página única (SPA).

## BCRYPT

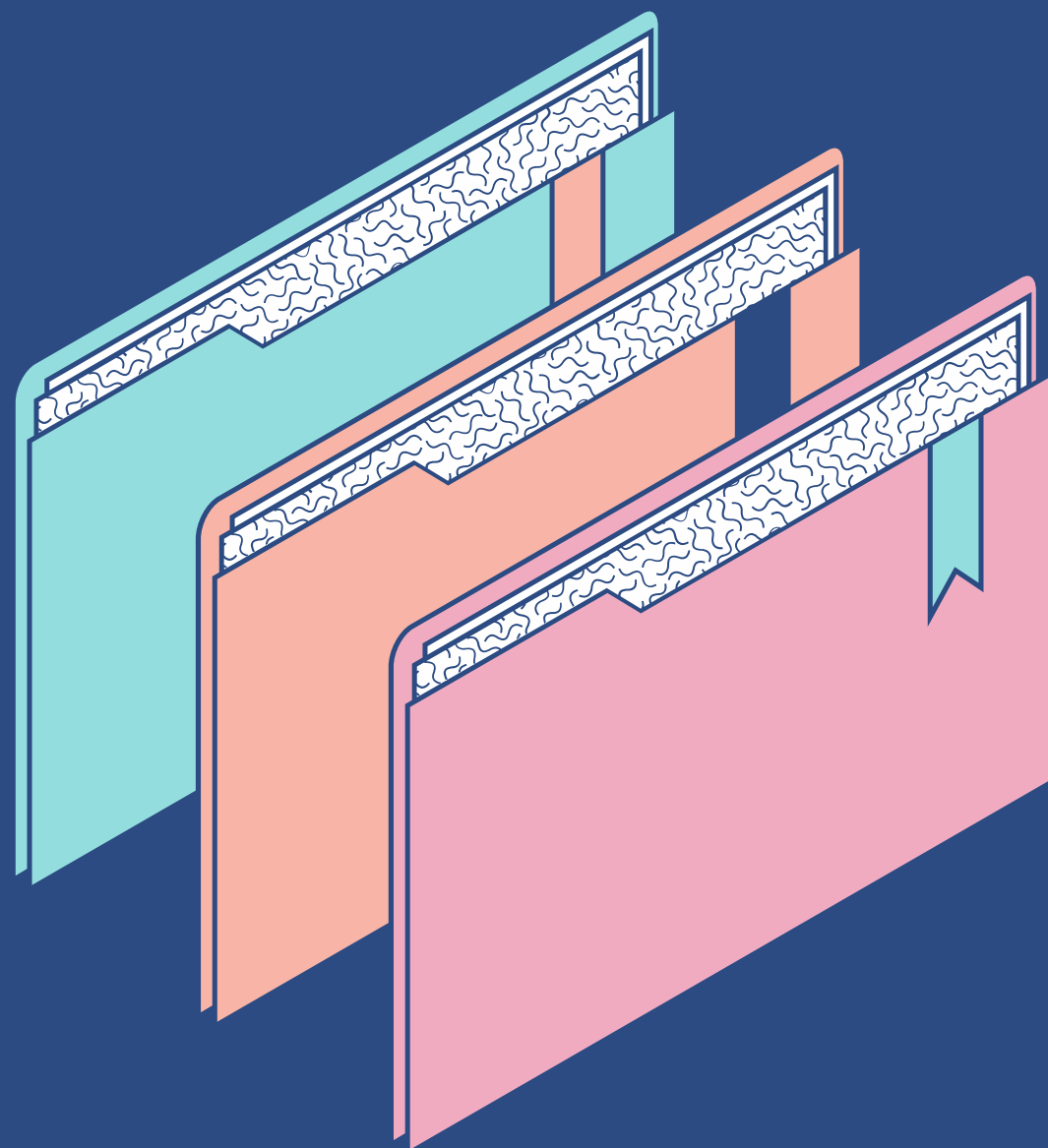
Uma biblioteca para criptografia de senhas em Node.js, comumente utilizada para proteger e armazenar senhas de forma segura em bancos de dados.

## CHAKRA UI

Um conjunto de componentes de interface de usuário React prontos para uso, permitindo a construção rápida e estilizada de interfaces com facilidade e consistência.

## MULTER

Pacote para Node.js que facilita o upload de arquivos, especialmente útil em aplicações que precisam lidar com o armazenamento e gerenciamento de uploads de imagens, vídeos, documentos, etc.



# Pasta Node Modules

Diretório fundamental em projetos Node.js, onde ficam armazenadas todas as dependências e pacotes instalados via gerenciador de pacotes, como o NPM. Ela contém todas as bibliotecas de terceiros necessárias para o projeto, facilitando a gestão das dependências. Esses módulos são baixados e mantidos localmente no diretório do projeto, permitindo que o código fonte utilize as funcionalidades desses pacotes de maneira modular e independente. Apesar de essencial, a pasta "node\_modules" pode ser bastante extensa, uma vez que pode conter diversos pacotes, mas seu conteúdo não é versionado no repositório de controle de código-fonte, para evitar ocupar espaço desnecessário e permitir que seja reconstruído facilmente em diferentes ambientes de desenvolvimento ou implantação.



# Banco de dados

Um banco de dados é um sistema organizado para armazenar, gerenciar e recuperar informações de forma estruturada. Funciona como um repositório onde dados são coletados e organizados de maneira lógica, permitindo o acesso, a manipulação e a consulta desses dados de forma eficiente. Esses sistemas são fundamentais para aplicações, pois oferecem a capacidade de persistir informações, facilitando a recuperação e modificação dos dados conforme necessário.

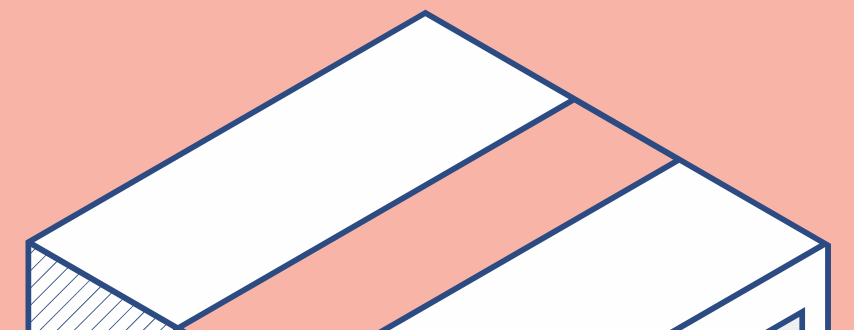
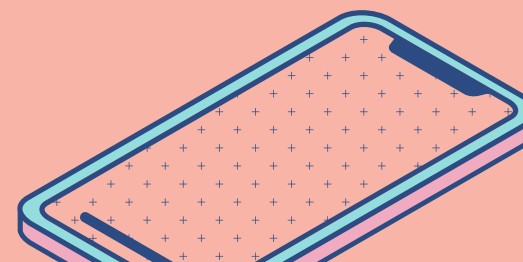
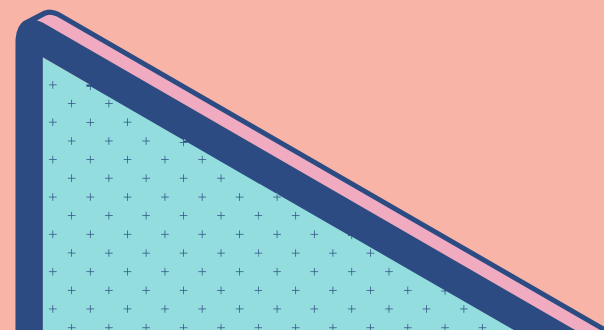
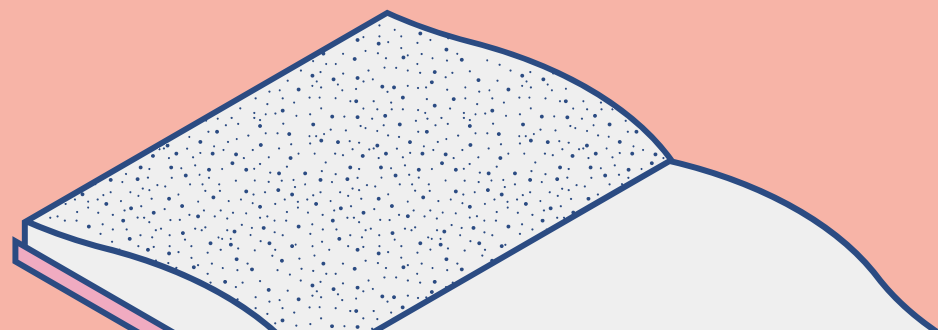
# MySQL

É um sistema de gerenciamento de banco de dados relacional amplamente utilizado no desenvolvimento com Node.js e Express.js. Como um dos sistemas de banco de dados relacionais mais populares, o MySQL oferece uma estrutura robusta para armazenar e recuperar dados de maneira eficiente, sendo frequentemente integrado a aplicações web desenvolvidas com Node.js. Sua escalabilidade, confiabilidade e desempenho o tornam uma escolha comum para muitos projetos, permitindo que desenvolvedores armazenem e gerenciem dados complexos de forma organizada e confiável, possibilitando a interação com o banco de dados por meio do Node.js com a ajuda de bibliotecas e drivers específicos.



# React.js

React desempenha um papel importante no desenvolvimento de interfaces de usuário eficientes e dinâmicas para aplicações web. Enquanto o Node.js é usado no backend para lidar com a lógica do servidor, o React é uma biblioteca JavaScript usada no frontend para criar interfaces de usuário interativas e componentizadas. A combinação do Node.js e React possibilita a construção de aplicações full-stack coesas, onde o Node.js atua como servidor e o React como a ferramenta principal para criar interfaces de usuário reativas, permitindo uma experiência de usuário fluida e responsiva. Esta integração harmoniosa entre backend e frontend proporciona um ambiente de desenvolvimento poderoso e coeso para a construção de aplicações modernas e escaláveis.



# VScode

O Visual Studio Code, é um editor de código-fonte altamente popular e poderoso desenvolvido pela Microsoft. Reconhecido por sua interface leve, personalizável e repleta de recursos, o VS Code é amplamente utilizado por desenvolvedores para programar em várias linguagens, incluindo JavaScript, TypeScript, Python, entre outras. Sua extensa gama de extensões, integração com controle de versão e depuração integrada tornam-no uma escolha favorita para o desenvolvimento de software, oferecendo uma experiência de desenvolvimento eficiente e adaptável às necessidades individuais de cada programador.





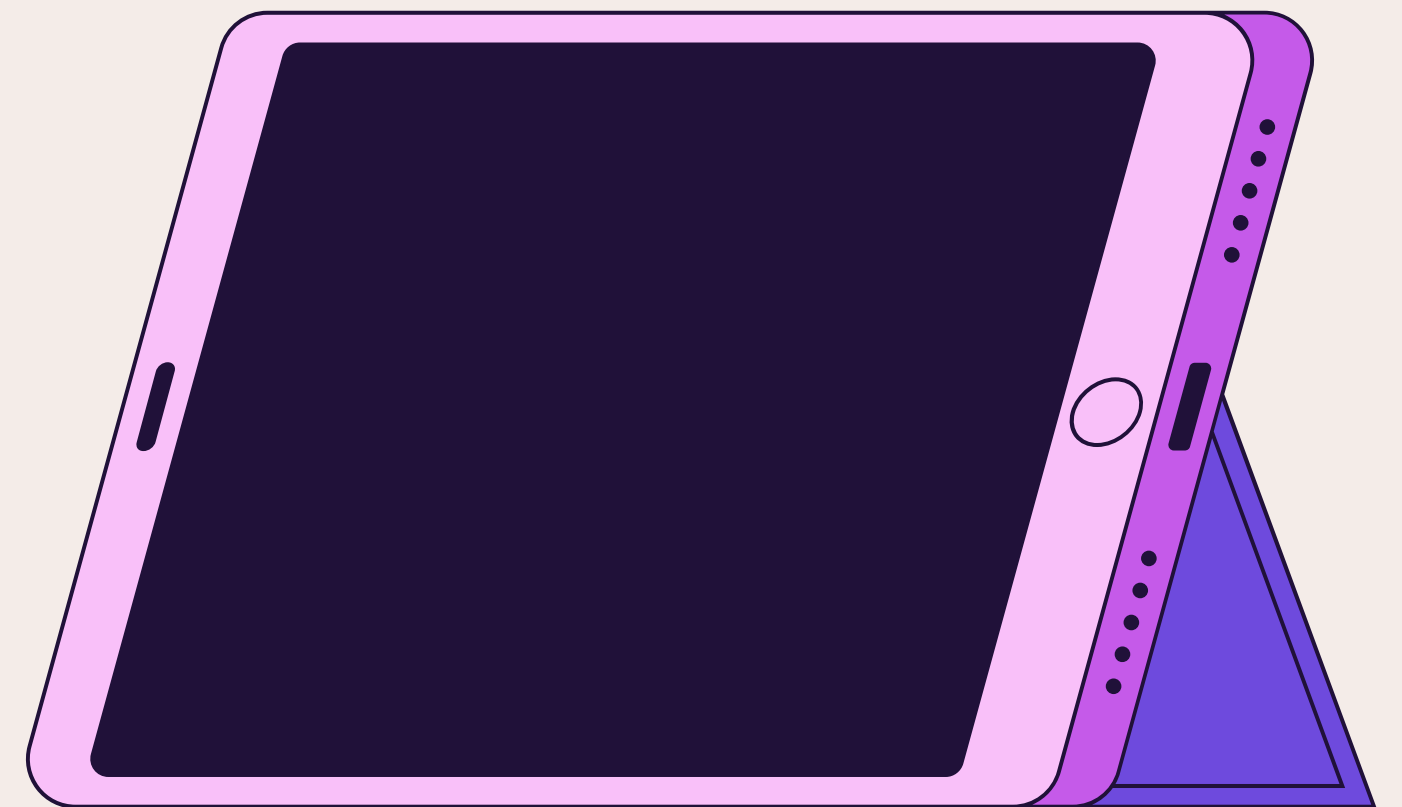
# O projeto

Sistema de anuncio de veículos

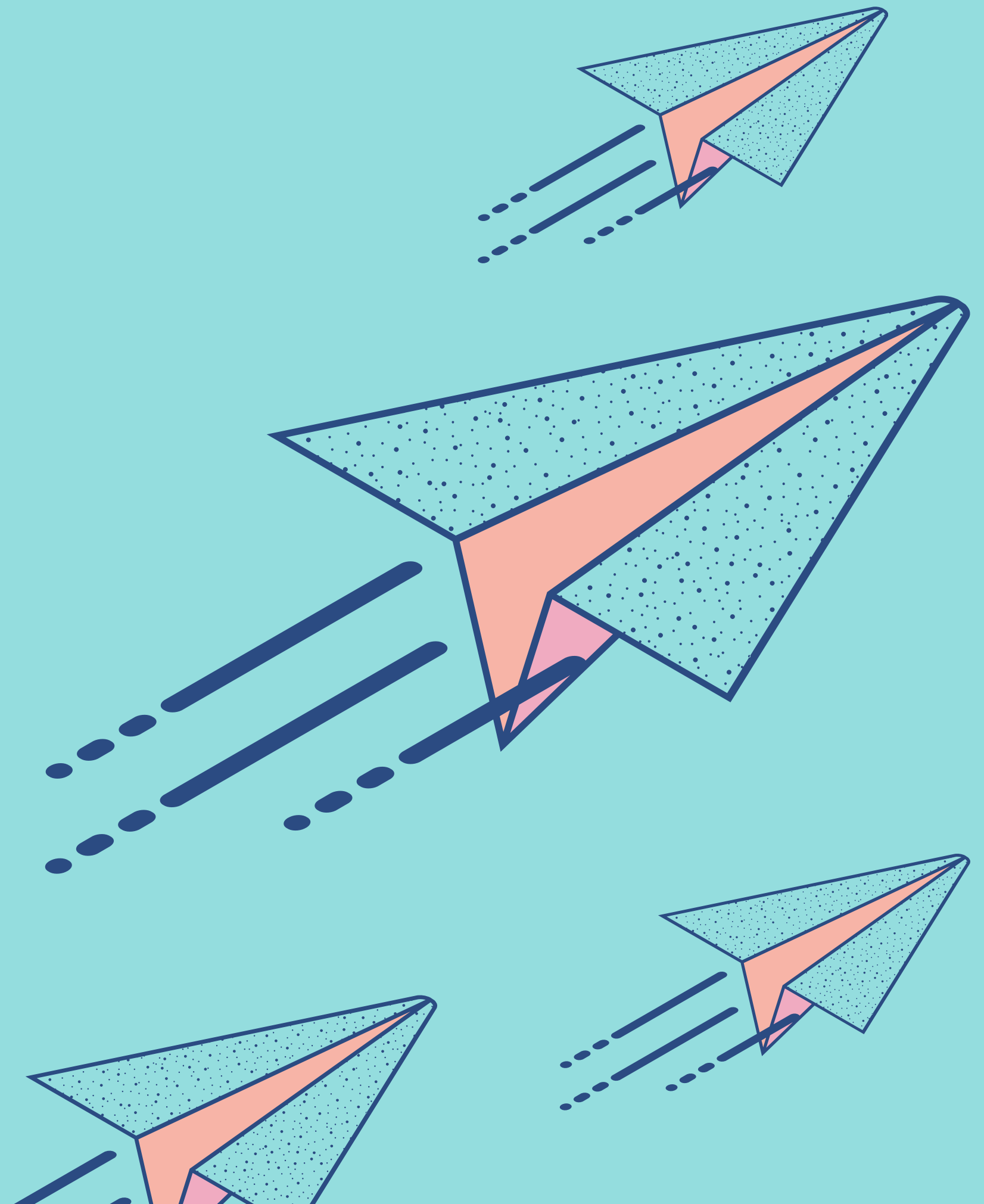
# Visão geral

Nosso projeto é um sistema de anúncios de veículos construído com tecnologias modernas. Utilizamos React.js em conjunto com a biblioteca Chakra UI para o front-end e Node.js em conjunto com Express no back-end. O banco de dados é gerenciado pelo MySQL, além de outras bibliotecas do Node.js para funcionalidades específicas.

O propósito desse sistema é proporcionar uma plataforma onde usuários possam facilmente anunciar veículos usados para venda. Ao mesmo tempo, oferece a possibilidade para outros usuários visualizarem esses anúncios e entrarem em contato para fechar negócio. A simplicidade e eficiência na interação entre compradores e vendedores são os pilares fundamentais desse projeto.



**Agora a  
apresentação do  
projeto**



# Configuração do Ambiente

**Itens necessários para a execução do projeto:**

**Node.js:** Baixar diretamente no site oficial do Node.

**MySQL:** Baixar diretamente no site do MySQL, junto com a ferramenta Workbench.

# Instalação das dependências

Express: `npm install express`

Nodemon: `npm install nodemon`

React: `npx create-react-app`

cors: `npm install cors`

bcrypt `npm install bcrypt`

multer `npm install multer`

# Execução do programa

Após a instalação das dependências, basta instalar o driver Mysql com o comando: *npm install mysql* e configurar o arquivo *db.js*, adicionando o nome de usuário e senha do seu banco de dados local.

Para executar o programa, basta acessar a pasta *server* no terminal e executar o comando: *npm start*. Após o servidor ser inicializado, basta acessar a pasta *client* no terminal e executar o comando: *npm start*. Com isso, o sistema estará funcionando localmente.