

# Programação Web com Linguagens de Script

Thiago Rodrigues



# Node.js

- O Node.js é um ambiente de execução de código JavaScript do lado do servidor, de código aberto e multiplataforma.
- **Características principais:**
  - **Assíncrono e orientado a eventos:** O Node.js usa um modelo assíncrono e orientado a eventos, o que significa que ele pode lidar com várias solicitações ao mesmo tempo sem bloquear.
  - **JavaScript do lado do servidor:** Com o Node.js, você pode usar JavaScript para escrever código do lado do servidor, como APIs da web, servidores de aplicativos e ferramentas de linha de comando.
  - **De código aberto e multiplataforma:** O Node.js é de código aberto e gratuito, e pode ser executado em uma variedade de sistemas operacionais, incluindo Windows, macOS e Linux.

# Express

- **Express.js**, também conhecido simplesmente como Express, é uma estrutura popular para a construção de aplicações web em Node.js. É conhecido por ser:
  - **Leve:** O Express fornece um conjunto essencial de recursos para lidar com a funcionalidade do servidor web sem ser intrusivo. Você pode personalizá-lo facilmente com bibliotecas adicionais.
  - **Flexível:** O Express não o força a usar ferramentas ou bibliotecas específicas. Você tem a liberdade de escolher o que funciona melhor para o seu projeto.
  - **Minimalista:** O Express se concentra em fornecer o essencial para a construção de aplicativos da web. Você pode adicionar funcionalidades como modelos ou acesso a banco de dados por meio de middleware.

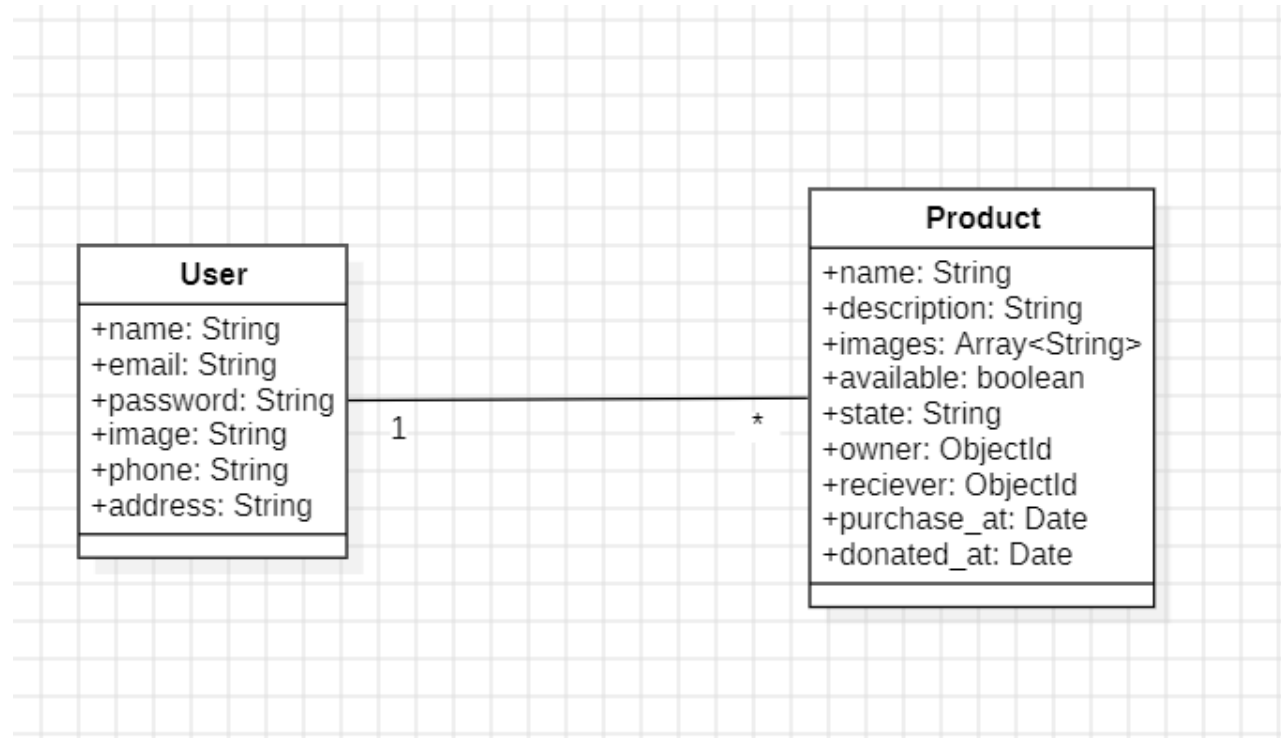
# Express

- **Aspecto Chave**

- **Roteamento:** O Express permite definir rotas para lidar com diferentes solicitações HTTP (GET, POST, etc.) para URLs específicas em seu aplicativo.
- **Middleware:** As funções de middleware são usadas para processar solicitações e respostas antes que cheguem ao manipulador final. Isso permite tarefas como registro, autenticação e análise de dados.

# Desenvolvimento da API

- Api para um Bazar
  - Diagrama de classe

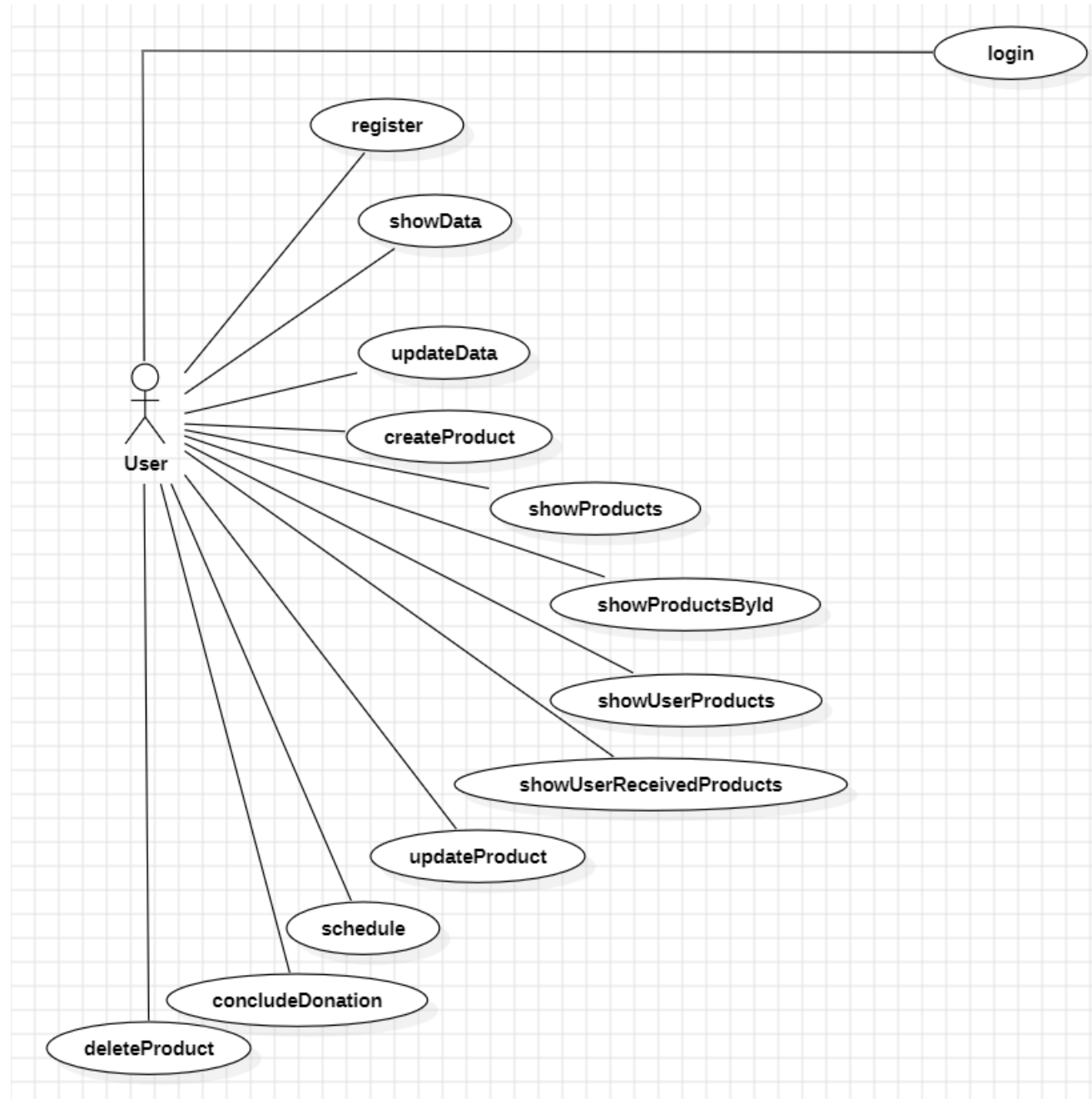


# Desenvolvimento da API

- **Funcionalidades**

- **Autenticáveis**

- showData
    - updateData
    - createProduct
    - showUserProducts
    - showUserReceivedProducts
    - updateProduct
    - schedule
    - concludeDonation
    - deleteProduct



# Arquitetura do Projeto

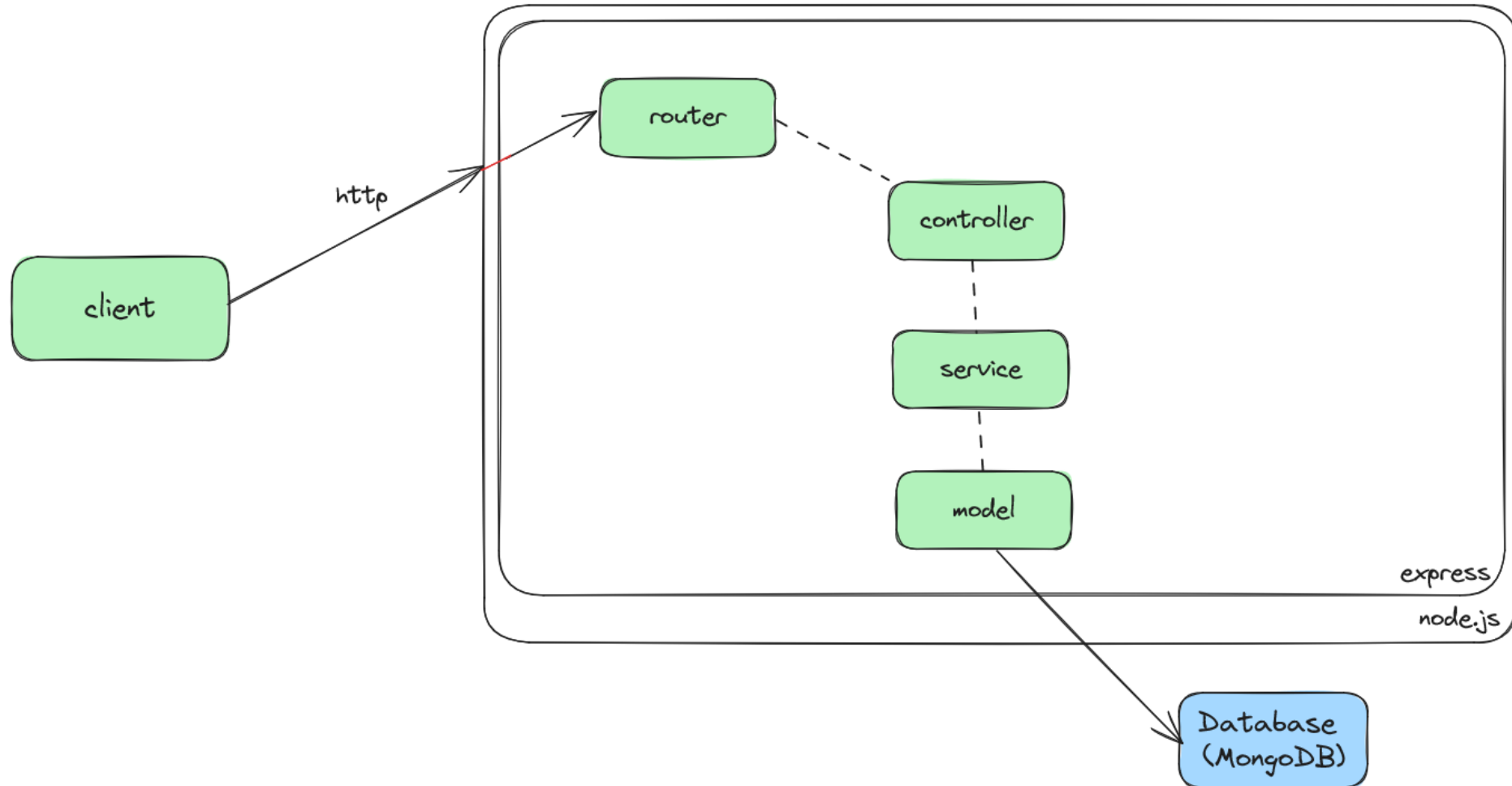
- **Arquitetura MVC**

- Um padrão de design para organizar aplicações em 3 camadas:
  - **Modelo:** Armazena e manipula dados.
  - **Visão:** Exibe dados para o usuário.
  - **Controlador:** Faz a ponte entre modelo e visão.

- **Camada de Serviço**

- Em um contexto de Arquitetura MVC (Modelo-Visão-Controlador), a **camada de serviço** (também conhecida como camada de negócio ou camada de aplicação) se destaca como um componente opcional, mas crucial para a organização e escalabilidade da aplicação.
- Essa camada atua como um **mediador entre as camadas de controlador e de modelo**, assumindo responsabilidades importantes na manipulação de dados e na aplicação de regras de negócio.

# Arquitetura do Projeto





# Gerenciados de dependência

- O **npm**, que significa **Node Package Manager**, é um gerenciador de pacotes crucial para o desenvolvimento com **Node.js**.
- Ele funciona como um repositório gigante de software, onde você encontra diversos módulos e bibliotecas pré-escritos para facilitar seu trabalho.
- **Principais funções do npm**
  - Instalar pacotes
  - Gerenciar dependências
  - Publicar pacotes
  - Gerenciar versões

# Dependências

- **Express**
- **Bcrypt**
  - O Bcrypt (Blowfish Crypt) é um algoritmo de hash usado para armazenar senhas de forma segura em aplicações web e desktop. Ele se destaca por sua **alta segurança e robustez contra ataques de força bruta**, tornando-o uma escolha popular para proteger dados sensíveis.
- **Multer**
  - O Multer é um **middleware popular para o Express.js** que facilita o upload de arquivos em aplicações web. Ele permite que você receba e processe arquivos enviados por usuários de forma simples e eficiente.

# Dependências

- **JsonWebToken**

- É um **pacote popular para o Node.js** que facilita a criação e validação de **tokens JSON Web (JWT)**. Esses tokens são utilizados para implementar **autenticação e autorização em aplicações web e APIs**.

- **Dotenv**

- O **dotenv** é uma ferramenta popular para gerenciar variáveis de ambiente em aplicações Node.js. Ele permite que você armazene informações de configuração confidenciais, como chaves de API, senhas e credenciais de banco de dados, em um arquivo separado chamado `.env` e, em seguida, carregue essas variáveis em seu código Node.js.

- **Nodemon**

- O Nodemon é uma **ferramenta de desenvolvimento para Node.js** que **monitora as mudanças em seus arquivos de código e reinicia automaticamente o servidor quando necessário**.

# Dependências

- **Mongoose**

- Mongoose, no contexto de desenvolvimento web, é uma biblioteca popular para Node.js que funciona como um **Object Data Modeling (ODM)**. Isso significa que ela facilita o processo de modelar seus dados de aplicação e interagir com um banco de dados MongoDB.