

**FECAP**



**PROJETO INTERDISCIPLINAR**

**GRUPO 11 - RECICLOTEC**

**Arquitetura da Implementação do Sistema.**

São Paulo

2024

**INTEGRANTES DO PROJETO e RA'S**

Alexandra Christine Silva	-	24026156
Gabrielly Cintra de Jesus	-	24025696
Hebert dos Reis Esteves	-	24026079
José Bento Almeida Gama	-	24026127

## **Sumário**

<b>1. INTRODUÇÃO .....</b>	<b>3</b>
<b>2. VISÃO GERAL DA ARQUITETURA .....</b>	<b>4</b>
<b>2.1 Fluxograma: Arquitetura da Implementação do Sistema - RecicloTec .</b>	<b>4</b>
<b>4. Tecnologias Utilizadas.....</b>	<b>6</b>
<b>5. Descrição das Camadas do Sistema .....</b>	<b>7</b>
<b>6. Fluxo de Funcionamento do Sistema .....</b>	<b>8</b>
<b>7. Conclusão .....</b>	<b>9</b>
<b>8. Referências .....</b>	<b>10</b>

## **1. INTRODUÇÃO**

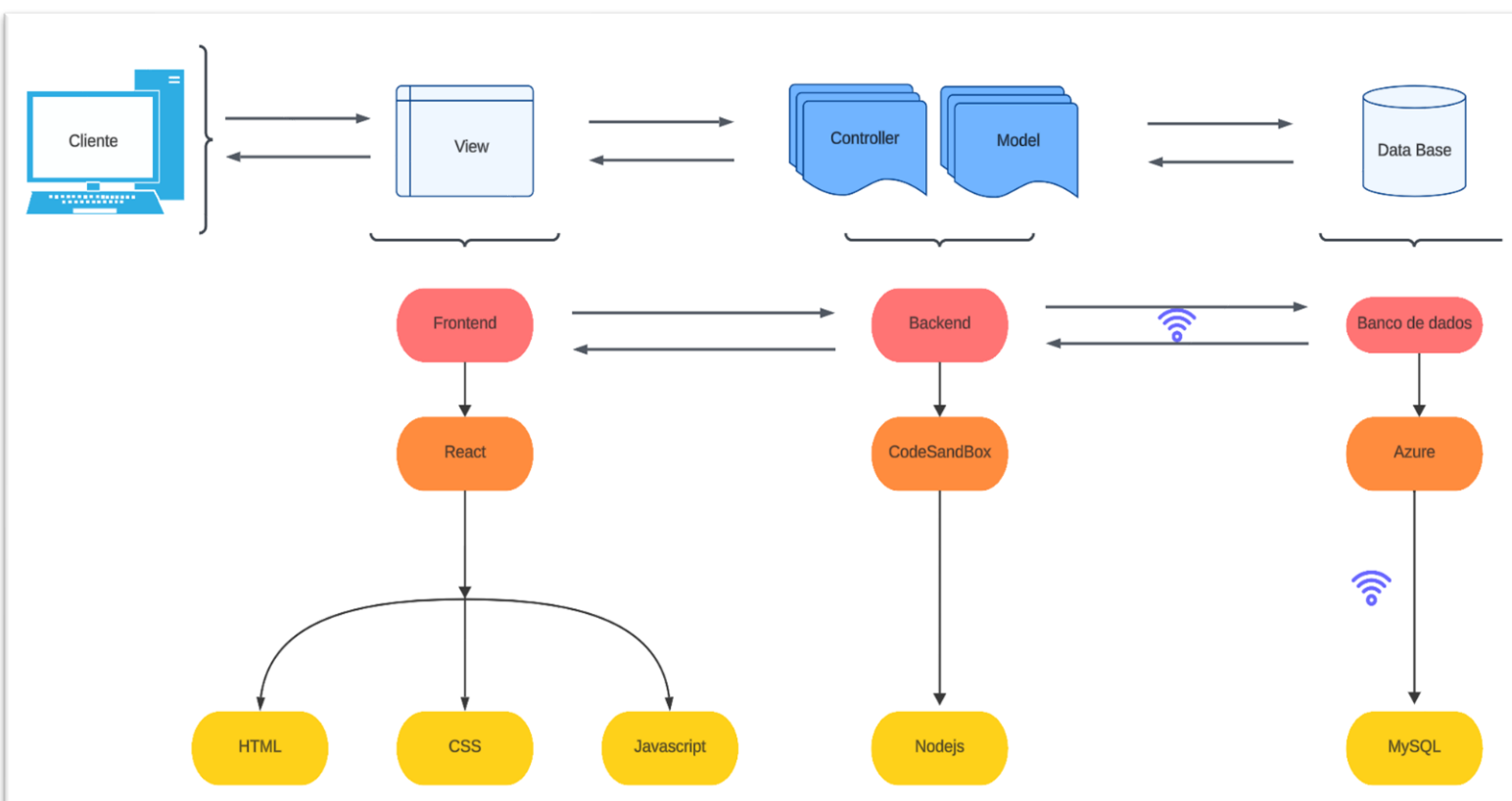
Este documento descreve a arquitetura do sistema do site ReCicloTec, que segue o padrão de desenvolvimento MVC (Model-View-Controller). Utilizando tecnologias modernas, o sistema visa representar a estrutura e arquitetura do site, garantindo uma experiência de uso fluida e segura para o agendamento de coleta de lixo eletrônico.

## 2. VISÃO GERAL DA ARQUITETURA

O sistema segue a estrutura MVC com uma clara separação entre frontend, backend e banco de dados. A aplicação é composta pelas seguintes camadas:

- **Frontend (Cliente e View):** Desenvolvido com React.
- **Backend (Controlador e Modelo):** Construído utilizando Node.js.
- **Banco de Dados:** Gerido em MySQL, hospedado na plataforma Azure.

### 2.1 Fluxograma: Arquitetura da Implementação do Sistema – RecicloTec



### 3. Diagrama de Arquitetura (Descrição do Diagrama)

- O diagrama ilustra a arquitetura do sistema em três partes principais:
- **Frontend (Cliente e View)** → Representado pelas tecnologias HTML, CSS, JavaScript e React.
- **Backend (Controller e Model)** → Desenvolvido em Node.js, com o suporte do ambiente CodeSandbox.
- **Banco de Dados** → Conectado ao backend via Azure, utilizando MySQL.

## 4. Tecnologias Utilizadas

### Front-end:

- **React:** Utilizado para criar interfaces dinâmicas.
- **HTML, CSS, JavaScript:** Para estruturação, estilização e interatividade da interface do usuário.

### Back-end:

- **Node.js:** Plataforma utilizada para desenvolver a camada de backend.
- **CodeSandbox:** Ambiente de desenvolvimento e teste para backend.

### Banco de Dados:

- **MySQL:** Sistema de banco de dados relacional para armazenar informações.
- **Azure:** Serviço em nuvem para hospedar o banco de dados, garantindo escalabilidade e segurança.

## 5. Descrição das Camadas do Sistema

### 5.1. Frontend (Usuário)

O frontend é responsável pela interação com o usuário final. A interface é construída utilizando React, proporcionando uma experiência fluida. A aplicação realiza operações de forma dinâmica.

### 5.2. Backend (Servidor)

A camada backend é desenvolvida em Node.js e é responsável pelo processamento das requisições recebidas do frontend, aplicação das regras de negócio e interação com o banco de dados.

**Processamento (Controller):** O **Controller** recebe como requisições HTTP do frontend, processa a lógica de negócio necessária e gerencia a comunicação com o banco de dados. Ele determina qual ação tomar, como enviar os dados para o **Model** ou responder ao **View**.



## 6. Fluxo de Funcionamento do Sistema

- **Usuário** acessa a aplicação web no navegador (Frontend).
- A interface (View) construída em **React** envia uma requisição ao backend (Node.js) quando o usuário realiza uma ação.  
O **backend (Controller)** processa a requisição, interage com o **Model** para acessar o **Banco de Dados (MySQL)** hospedado no **Azure**.
- O backend retorna os dados processados para o frontend, que atualiza a interface para o usuário em tempo real.

## 7. Conclusão

A arquitetura descrita promove uma separação clara entre as camadas, garantindo que o sistema seja:

- **Escalável:** Facilmente expansível para atender a um maior número de usuários.
- **Modular:** Manutenção simplificada devido à separação de responsabilidades.
- **Eficiente:** Utilização de tecnologias modernas como React e Node.js para alto desempenho.

## 8. Referências

1. SILVA, Mayk Brito da. **Arquitetura MVC: entendendo o modelo de visão driver** . DIO, 2023. Disponível em: <https://www.dio.me/articles/arquitetura-mvc-entendendo-o-modelo-visao-controlador> . Acesso em: 17 nov. 2024.
2. COODESH. **O que é Arquitetura MVC?** Coodesh, 2023. Disponível em: <https://coodesh.com/blog/dicionario/o-que-e-arquitetura-mvc/> . Acesso em: 17 nov. 2024.
3. LE WAGON. **O que é o padrão MVC?** Le Wagon Blog, 2023. Disponível em: <https://blog.lewagon.com/pt-br/skills/o-que-e-padrao-mvc/> . Acesso em: 17 nov. 2024.