



Universidade de Fortaleza
Centro de Ciências Tecnológicas
Curso de Ciência da Computação
Disciplina Ambiente de Dados

Documento Explicativo
Sistema de Delivery - React e Node.js

Equipe:
João Pedro Soares Franco - 2419458
Lara Stephanny Lima Gomes - 2410494

Fortaleza, 2025

1. Funcionamento do Sistema

1.1 Objetivo

O sistema desenvolvido é uma aplicação Full Stack para gerenciamento de um serviço de Delivery de comida. O objetivo é permitir o controle completo do fluxo de pedidos, desde o cadastro dos envolvidos até a entrega final. O sistema realiza as operações de CRUD (Create, Read, Update, Delete) para gerenciar Restaurantes, Clientes e Pedidos, além de controlar o status da entrega em tempo real.

1.2 Tecnologias Utilizadas

Para o desenvolvimento deste projeto, foram utilizadas tecnologias modernas baseadas em JavaScript:

- **Runtime:** Node.js (Versão LTS).
- **Back-End:** Express (Framework para API), MySQL2 (Driver de conexão), Cors (Controle de acesso), Dotenv (Variáveis de ambiente) e Nodemon (Atualização automática do servidor).
- **Front-End:** React (Biblioteca de interfaces), Vite (Build tool) e Axios (Consumo de API).
- **Estilização:** Tailwind CSS (Framework de CSS utilitário).
- **Banco de Dados:** MySQL 8.0.
- **Ferramentas de Desenvolvimento:** Visual Studio Code (Ambiente de desenvolvimento) e Postman (Ferramenta para testes de rotas API).

1.3 Arquitetura do Projeto

O projeto foi estruturado utilizando o padrão **MVC (Model-View-Controller)** adaptado para uma API REST, visando a separação de responsabilidades e facilitando a manutenção:

- **Model:** Responsável pela comunicação direta com o Banco de Dados (SQL). Contém as funções que executam INSERT, SELECT, UPDATE e DELETE nas tabelas do MySQL.
- **Controller:** É o intermediário. Ele recebe as requisições do Front-End, valida os dados (regras de negócio) e chama o Model para executar a ação.
- **Routes:** Define os "caminhos" (URLs) da API que o Front-End pode acessar (ex: /pedido, /cliente).
- **View (Front-End):** Desenvolvida em React, é a camada visual onde o usuário interage, preenche formulários e visualiza os dados retornados pelo Back-End.

2. Guia para Reprodução (Passo a passo)

Este guia permite que qualquer desenvolvedor replique e execute o projeto do zero, desde a instalação do ambiente.

Passo 1: Instalação do Ambiente de Desenvolvimento

Antes de executar o código, precisamos preparar o computador.

2.1.1 Instalação do Node.js

O Node.js é o ambiente de execução necessário para rodar tanto o servidor quanto o ambiente de desenvolvimento do React.

1. Acesse o site oficial: nodejs.org
2. Baixe a versão **LTS** (Long Term Support) recomendada para a maioria dos usuários.
3. Execute o instalador e siga o assistente (Next > Next > Install).
4. **Verificação:** Abra o terminal (CMD ou PowerShell) e digite node -v. Deve aparecer a versão instalada.

2.1.2 Instalação do MySQL Server e Workbench

Usaremos o MySQL para armazenar os dados.

1. Acesse o site oficial do MySQL: dev.mysql.com/downloads/installer/
2. Baixe o "MySQL Installer for Windows".
3. Execute o instalador e escolha a opção "**Developer Default**".
4. **Importante:** Durante a configuração de "Accounts and Roles", defina uma senha para o usuário root e **anote-a**. Você precisará dela para conectar o sistema.
5. Continue clicando em “next” e “execute” até que a instalação de todos os produtos esteja completa.

2.1.3 Instalação do VS Code (IDE)

Vamos utilizar o Visual Studio Code como Ambiente de Desenvolvimento

1. Acesse o site oficial: code.visualstudio.com/
2. Baixe e execute o instalador para o seu sistema operacional.
3. Siga o assistente de instalação
4. Após finalizar a instalação, abra o arquivo do projeto no Visual Studio Code

2.1.4 Instalação do Postman

O Postman é utilizado para testar e documentar as rotas da API (Back-End) antes de integrá-las ao Front-End.

1. Acesse o site oficial: postman.com/downloads
2. Baixe a versão compatível com seu sistema operacional
3. Instale e abra o aplicativo
4. Não é obrigatório criar conta para usar o básico, você pode optar por pular o login se preferir

Passo 2: Criação do Banco de Dados (Script SQL)

Agora vamos criar a estrutura do banco de dados necessária para o sistema funcionar.

1. Abra o **MySQL Workbench**.
2. Conecte-se ao seu servidor local (Local instance) usando a senha root definida na instalação.
3. Abra uma nova janela de Query (ícone de arquivo com SQL ou File > New Query Tab).
4. Entre nesse link: <https://github.com/jotapz/sistema-delivery>
5. Vá no arquivo delivery.sql copie ou baixe
6. Se tiver baixado abra o script e execute o código clicando no ícone do raio
7. Se tiver copiado, cole o script e execute o código selecionando as linhas abaixo e clicando no raio.

- **CREATE DATABASE delivery;**
- **USE delivery;**
- **CREATE TABLE Cliente();**
- **CREATE TABLE Restaurante();**
- **CREATE TABLE Pedido();**
- **CREATE TABLE ItemPedido();**
- **CREATE TABLE Funcionario();**

Ao final, você terá o banco delivery criado com as 5 tabelas.

Passo 3: Configuração e Execução do Back-End

O Back-End é o servidor que conecta o banco de dados à interface.

- 1.** Inicie o Visual Studio Code
- 2.** Abra o terminal do Visual Studio Code
- 3.** Entre na pasta backdelivery com o comando: **cd backdelivery**
- 4.** Dentro da pasta você terá que instalar as dependências do projeto com o comando: **npm install** (Isso baixará as bibliotecas **express**, **mysql2**, **cors**, **dotenv** e **nodemon**)
- 5.** Configuração de segurança: Crie um arquivo chamada **.env** na raiz da pasta **backdelivery** e insira suas credenciais do banco:
 - DB_HOST=localhost
 - DB_USER=root
 - DB_PASSWORD=sua_senha_do_mysql (**senha criada na instalacao**)
 - DB_DATABASE=delivery
 - PORT=5001
- 6.** Execute o servidor utilizando o comando: **npm run dev**

Se tudo estiver correto, aparecerá a mensagem: "Servidor rodando na porta 5001".

Passo 4: Configuração e Execução do Front-End

O Front-End é a interface visual do sistema.

- 1.** Abra um novo terminal no Visual Studio Code (**deixe o backdelivery rodando**)
- 2.** Entre na pasta utilizando o comando: **cd frontdelivery**
- 3.** Instale as dependências (React, Tailwind, Axios) utilizando o comando: **npm install**
- 4.** Execute o front com o comando: **npm run dev**
- 5.** O terminal mostrará um link que quando você clicar, será redirecionado para a página do front no navegador

3. Manual de Uso e Funcionalidades

Esta seção descreve como utilizar o sistema após a sua execução.

3.1 Configuração inicial de acesso (Criar Funcionário)

Antes de tentar fazer login, é necessário cadastrar primeiro o funcionário no banco de dados. Existem duas formas de fazer isso:

- 1. Via MySQL Workbench:** Dentro do banco, use o banco de dados delivery e execute o comando **INSERT INTO Funcionario (nome, email, senha) VALUES ('Admin', 'admin@delivery', '123');**
 - *Nao é necessário os dados serem exatamente esses, você pode colocar seu nome, email e senha se quiser*

- 2. Via Postman:** Serve para testar a API, então você pode criar por lá também assim:

1. Abra o Postman.
2. Crie uma requisição do tipo POST
3. URL: <http://localhost:5001/funcionario/registrar>
4. No corpo Body -> Raw -> JSON, insira os dados em formato JSON

The screenshot shows the Postman interface with a POST request to `localhost:5001/funcionario/registrar`. The request body is set to `raw` and contains the following JSON:

```
1 {  
2   "nome": "joao",  
3   "email": "jpedro@gmail.com",  
4   "senha": "12345"  
5 }
```

The response section shows a `201 Created` status with a response body containing:

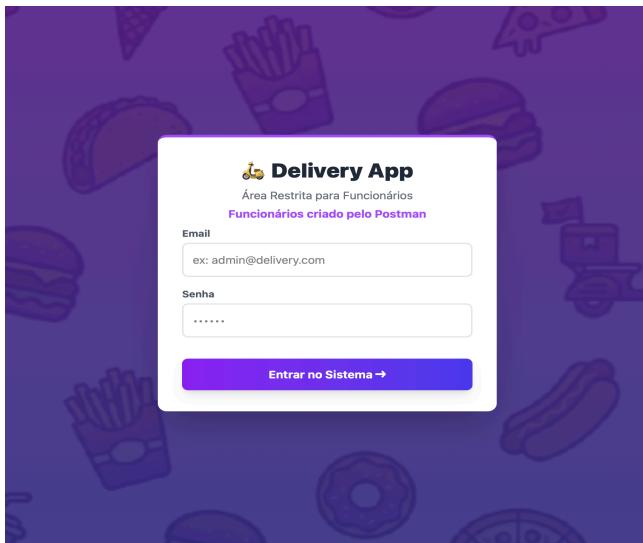
```
1 {  
2   "message": "Funcionário cadastrado com sucesso!",  
3   "id": 2  
4 }
```

5. Clique em Send, se retornar o Status 201 o usuário foi criado.

3.2 Acesso ao Sistema (Login)

Ao abrir a aplicação no navegador, você será direcionado a tela de Login.

1. Insira o email que você cadastrou no passo anterior.
2. Insira a senha que você cadastrou.
3. Clique em “Entrar no Sistema”.

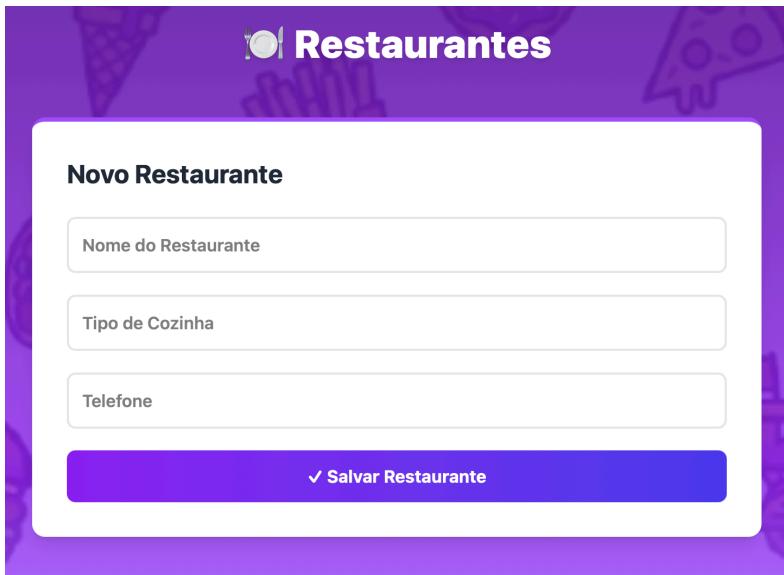


3.3 Cadastros (Restaurantes e Clientes)

Com o acesso liberado, você pode navegar pelo sistema.

1. Restaurantes:

- Acesse a aba "Restaurantes".
- Preencha Nome, Tipo de Cozinha e Telefone.
- Clique em "Adicionar". O restaurante aparecerá na lista abaixo.

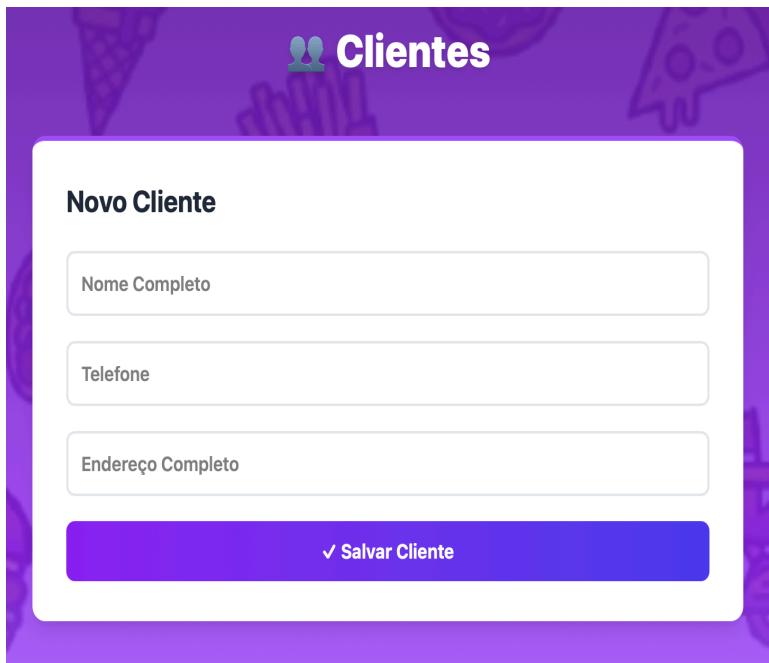


2. Clientes:

- Acesse a aba "Clientes".

- Preencha Nome, Telefone e Endereço.

- Clique em "Salvar Cliente".



3.4 Realização de Pedidos

Este é o módulo principal do sistema.

1. Acesse a aba "Fazer Pedido".
2. Escolha um Cliente e um Restaurante nos menus de seleção (listados diretamente do banco de dados).
3. Montar Carrinho: Digite a descrição do item (ex: "Pizza"), a quantidade e o preço. Clique em + Add.
4. Repita o processo para adicionar mais itens ao mesmo pedido.
5. Clique no botão verde "**Finalizar Pedido**". O sistema salvará o pedido e os itens vinculados no banco de dados.



3.3 Gerenciamento de Status

Após criar o pedido, ele deve ser gerenciado na lista "Pedidos em Andamento" (na parte inferior da tela de Pedidos).

1. O pedido nasce com o status Em Preparo (Cor Amarela).
2. Ao clicar no botão "**Mandar Entregar**", o status muda para A Caminho (Cor Azul).
3. Ao clicar no botão "**Confirmar Entrega**", o status muda para Entregue (Cor Verde), finalizando o processo.

4. Reppositórios do Projeto (GitHub)

Abaixo encontram-se os links para acesso ao código-fonte dos projetos desenvolvidos na disciplina.

4.1 Projeto Atual: Sistema de Delivery Full Stack

Tecnologias: React, Node.js, MySQL.

Link do Repositório: <https://github.com/jotapz/sistema-delivery>

4.2 Projeto Anterior: Sistema de Gestão de Academia

Tecnologias: Java, Swing, JDBC, MySQL.

Link do Repositório: <https://github.com/jotapz/gestao-academia>