Guia de Configuração de Ambiente de Desenvolvimento

Feito por :

Nicolas dos Santos Xavier - 2320299

João Lucas Lima Maia - 2410532

1. Objetivo do Documento

Este documento fornece um guia completo e detalhado para configurar o ambiente de desenvolvimento do projeto **ambienteDeDados**, permitindo que novos desenvolvedores possam configurar, executar e testar a API de forma rápida e eficiente.

2. Público-Alvo

Este guia destina-se a membros da equipe de desenvolvimento, novos colaboradores e qualquer desenvolvedor que necessite configurar o ambiente local para contribuir com o projeto.

3. Stack de Tecnologias (Especificações Técnicas)

Com base no package.json do projeto, estas são as tecnologias e bibliotecas utilizadas:

3.1 Núcleo e Servidor

- **Linguagem:** JavaScript.
- **Ambiente de Execução:** Node.js (Versão 18 ou superior recomendada).
- **Framework Principal:** express - Utilizado para criar o servidor web e gerenciar as rotas da API.

3.2 Banco de Dados

- **SGBD:** MySQL.
- **Driver:** mysql2 - Responsável pela conexão e execução de queries no banco de dados.

3.3 Segurança e Autenticação

- **Tokens de Acesso:** jsonwebtoken - Utilizado para gerar e validar JWTs, protegendo rotas autenticadas.
- **Criptografia:** bcryptjs - Utilizado para realizar o hash de senhas antes do armazenamento.
- **Acessibilidade (CORS):** cors - Gerencia o compartilhamento de recursos entre origens diferentes (essencial para integração com Frontends).

3.4 Configuração e Ambiente

- **Variáveis de Ambiente:** dotenv - Carrega dados sensíveis (credenciais, secrets) a partir do arquivo .env.
- **Scripts Multiplataforma:** cross-env - Garante que variáveis como NODE_ENV sejam definidas corretamente independente do sistema operacional (Windows/Linux/macOS).

3.5 Desenvolvimento e Testes

- **Hot-Reload:** nodemon - Monitora alterações nos arquivos e reinicia o servidor automaticamente durante o desenvolvimento.
- **Testes:** jest (Framework de testes) e supertest (para requisições HTTP nos testes de integração).

---

4. Como Configurar e Rodar o Projeto (Passo a Passo)

Siga rigorosamente as etapas abaixo para configurar o ambiente local.

Passo 1: Pré-requisitos [4]

Garanta que você tenha os seguintes softwares instalados:

- **Node.js** (v18 ou superior).
- **Git** (para clonagem do repositório).
- **MySQL Server** (Instalado localmente ou rodando via Docker).

Passo 2: Clonar e Instalar Dependências

Abra seu terminal e clone o repositório oficial:
Bash

```
git clone https://github.com/Nicolas/ambienteDeDados.git
```

Navegue até a pasta do serviço Node.js:
Bash

```
cd ambienteDeDados/node.js
```

Instale todas as dependências listadas no package.json:
Bash

```
npm install
```

Passo 3: Configurar Variáveis de Ambiente

1. Na raiz da pasta node.js, crie um arquivo chamado .env.

Copie o conteúdo abaixo e ajuste os valores conforme suas configurações locais (usuário e senha do MySQL):
Ini, TOML
Configurações do Servidor
PORT=3333

Configurações do Banco de Dados MySQL
DB_HOST=localhost
DB_USER=seu_usuario_mysql # <--- ALTERE AQUI
DB_PASSWORD=sua_senha_mysql # <--- ALTERE AQUI
DB_DATABASE=nome_do_seu_banco # <--- ALTERE AQUI

Segredo para o JWT (JSON Web Token)
JWT_SECRET=seu_segredo_super_secreto_aqui

2. **Importante:** Nunca commite o arquivo .env no Git. Ele contém dados sensíveis.

Passo 4: Preparar o Banco de Dados

1. Certifique-se de que o serviço MySQL está rodando.
2. Crie o banco de dados (Schema) com o nome definido em DB_DATABASE no seu arquivo .env.
3. Execute os scripts de criação de tabelas/migrações (se disponíveis no projeto) para estruturar o banco.

Passo 5: Executar a Aplicação
O projeto possui scripts configurados no package.json para facilitar a execução:

Para Desenvolvimento (Recomendado):
Utiliza o nodemon para reiniciar o servidor a cada salvamento de arquivo.
Bash

```
npm run dev
```

Para Produção/Execução Simples:
Roda a aplicação diretamente com o node.
Bash

```
npm run start
```

Sucesso: O terminal deverá exibir: Servidor rodando na porta 3333.

Passo 6: Executar os Testes
Para garantir a integridade da API e verificar os endpoints:

Bash
npm test

Este comando acionará o jest para rodar todos os arquivos *.test.js ou *.spec.js.

---

5. Solução de Problemas Comuns (Troubleshooting)

Problema	Causa Provável	Solução
Erro: connect ECONNREFUSED	O MySQL não está rodando ou a porta está errada.	Verifique se o serviço do MySQL está ativo e se a porta no .env está correta (padrão 3306).
Erro: Access denied for user	Credenciais do banco incorretas.	Verifique DB_USER e DB_PASSWORD no arquivo .env.
Comando 'npm' não encontrado	Node.js não instalado ou não está no PATH.	Reinstale o Node.js e marque a opção "Add to PATH" durante a instalação.
Erro: EADDRINUSE: address already in use	A porta 3333 já está sendo usada.	Altere a variável PORT no arquivo .env para outro valor (ex: 3334).
Módulos não encontrados (Error: Cannot find module)	Dependências não instaladas.	Execute npm install novamente na pasta raiz do projeto.

---

6. Boas Práticas de Desenvolvimento

1. **Segurança:** Nunca exponha o JWT_SECRET ou senhas do banco em repositórios públicos. Use sempre o .env.
2. **Padrão de Código:** Utilize recursos modernos do JavaScript (ES6+) e mantenha o código limpo.
3. **Testes:** Sempre rode npm test antes de enviar alterações para o repositório remoto para evitar quebrar funcionalidades existentes.

4. **Versionamento:** Mantenha o arquivo package-lock.json versionado para garantir que todos os desenvolvedores usem as mesmas versões das dependências.