

Nome do grupo: Grupo 3 – Rafael Justino, Guilherme de Andrade, Grazielle Miranda e Kauan Sano

Disciplina AS65A - Certificadora de Competência Identitária – Turma n15

Sumário

- Pré-requisitos e dependências.
- Configuração do Banco de Dados para vinculá-lo ao projeto.
- Configuração do projeto e acesso a rotas para popular o banco de dados.
- Execução e configuração do front e back-end.

1. Pré-requisitos

- Instalação do MySQL Server e MySQL Workbench
- Instalação do Visual Studio Code
- Instalação do Node.js
- Postman API Tester

Essas ferramentas foram utilizadas no desenvolvimento do projeto. Abaixo estão instruções detalhadas para a configuração do ambiente para executar a aplicação. Os arquivos necessários (arquivo .env, a coleção de testes no Postman, etc) estão disponíveis como arquivos no repositório.

Segue os links e versões para as ferramentas utilizadas:

VISUAL STUDIO CODE: <https://code.visualstudio.com> – Última versão estável disponibilizada.

MySQL Server: <https://dev.mysql.com/downloads/mysql/> - Versão mais recente disponibilizada (8.2.0).

MySQL Workbench: <https://dev.mysql.com/downloads/workbench/> - Versão disponibilizada neste link (8.0.34).

Node.js: <https://nodejs.org/en> - Versão 20.10.0 LTS.

Postman API Tester: <https://www.postman.com/downloads/> - Baixar e executar instalador disponibilizado.

2. Dependências e bibliotecas:

Todas estas dependências serão instaladas ao executar o comando NPM Install na pasta do projeto com o Visual Studio Code aberto. Aqui está uma relação destas:

CORS	2.8.5	https://www.npmjs.com/package/cors
dotenv	16.3.1	https://www.npmjs.com/package/dotenv
express	4.18.2	https://www.npmjs.com/package/express
express-validator	7.0.1	https://www.npmjs.com/package/express-validator
mysql2	3.6.1	https://www.npmjs.com/package/mysql2
nodemon	3.0.1	https://www.npmjs.com/package/nodemon
sequelize	6.33.0	https://www.npmjs.com/package/sequelize
swagger-jsdoc	6.2.8	https://www.npmjs.com/package/swagger-jsdoc

swagger-ui-express	5.0.0	https://www.npmjs.com/package/swagger-ui-express
Axios	1.6.0	https://www.npmjs.com/package/axios : https://www.npmjs.com/package/axios
eslint	8.49.0	https://www.npmjs.com/package/eslint : https://www.npmjs.com/package/eslint
eslint-config-next	13.5.2	https://www.npmjs.com/package/eslint-config-next : https://www.npmjs.com/package/eslint-config-next
react	18.2.0	https://www.npmjs.com/package/react : https://www.npmjs.com/package/react
react-dom	18.2.0	https://www.npmjs.com/package/react-dom : https://www.npmjs.com/package/react-dom
@types/node	20.8.9	https://www.npmjs.com/package/@types/node : https://www.npmjs.com/package/@types/node
@types/react	18.2.33	https://www.npmjs.com/package/@types/react : https://www.npmjs.com/package/@types/react

3. Configuração do Projeto:

3.1 - Banco de dados

Para o armazenamento de dados foi utilizado o banco de dados MySQL Workbench.

- Para vincular o projeto ao seu banco de dados MySQL, primeiro abra o MySQL e crie um schema com o nome "projetoidentitaria";
- Em seguida, abra o projeto "AS63A-IDENTITARIA-GRUPO03-BACK" e vá até o arquivo .env;

3.1.1 – Configuração do arquivo dotenv:

Dentro do arquivo .env, você encontrará os campos para configurar o projeto com seu banco de dados, atualize os dados conforme abaixo:

- DB_DIALECT -> Coloque "mysql";
- DB_HOST -> Coloque 127.0.0.1, este é endereço padrão utilizado pelo MySQL;
- DB_NAME -> Nome do schema que criou no banco de dados, se você seguiu os passos corretamente, o nome deve ser "projetoidentitaria";
- DB_PORT -> Coloque 3306, esta é a porta padrão do MySQL;
- DB_USER -> Coloque seu usuário do MySQL (o padrão quando se instala o MySQL é root);

- DB_PASSWORD -> coloque a senha do seu banco de dados, caso não tenha definido uma senha, deixe em branco;

Após realizar estas alterações, o projeto está pronto para ser vinculado ao seu banco de dados.

3.2 - Configuração do projeto: BACK-END

Após instalar as ferramentas necessárias e fazer o download do .zip do projeto disponibilizado pelo Github, é necessário seguir esses passos:

1. Descompactar o projeto em uma pasta de sua escolha.
2. Abrir o Visual Studio Code, e apertar os botões (CTRL + K) e depois (CTRL + O). Isso deve abrir uma janela que permitirá você navegar pelos seus arquivos e encontrar a pasta onde descompactou o projeto para selecioná-la.
3. Caso o atalho não funcione, navegue com o mouse até Arquivos (no canto superior da aplicação, ao lado do logotipo) e selecione a opção no dropdown para abrir uma pasta.
4. Com a pasta BACK-END (**AS65A-Identitaria-Grupo03-Back**) do projeto aberta dentro do Visual Studio Code, aperte (CTRL + SHIFT + `). Caso não funcione, navegue até a opção de Terminal -> Abrir um novo terminal.
5. Digite e execute o comando `node --version`. Deverá mostrar a versão do Node instalada na sua máquina. Caso o sistema não reconheça o comando, reinstale o Node.js seguindo algum material mais detalhado.
6. Assumindo que o Node.js está configurado na sua máquina, execute o comando `"npm install"`, sem as aspas. Aguarde e espere o sistema baixar os arquivos do projeto.
7. Após efetuar o download dos arquivos necessários, insira o comando: `"npm start"`. Se o banco de dados foi configurado corretamente, deverá exibir a seguinte mensagem no console: Conectado no MySQL!
8. O projeto está rodando na porta 3001. Para popular o banco de dados, abra um navegador de internet da sua escolha e navegue até o seguinte link: <http://localhost:3001/install>

Pronto! Após seguir esses passos, o banco de dados do projeto está populado e pronto para ser manipulado. Existem várias rotas em uso nessa aplicação, e todas elas se encontram no final desse documento na forma de uma coleção para o aplicativo Postman, que facilita o uso de rotas de cadastro e atualização. Agora, vamos configurar o Front-end:

3.3 - Configuração do projeto: FRONT-END

Após popular o banco de dados, é necessário configurar o front-end do projeto. Para seguir estes passos, abra a pasta **AS65A-Identitaria-Grupo03-Front** pelo Visual Studio Code. Assumindo que os passos anteriores foram realizados e a pasta está aberta no VSCode, siga estes passos:

1. Aperte (CTRL + SHIFT + '). Caso não funcione, navegue até a opção de Terminal -> Abrir um novo terminal.
2. Com o terminal aberto, execute o comando "npm install", sem as aspas. Aguarde o download das dependências do projeto.
3. Após as dependências serem instaladas, execute o comando "npm run dev". Ele deve disponibilizar uma mensagem de sucesso na tela e o link para acessar o projeto.
4. Acesse o link: <http://localhost:3000>
5. Antes de fazer o login, certifique-se que o back-end do projeto está em execução, conforme o passo anterior.
6. Acesse a aplicação com o usuário e senha: pedro123
7. Esse usuário é um professor, que pode criar, alterar e excluir outros usuários, conforme foi definida nas regras e especificações de negócio do projeto.

Pronto! O projeto está configurado e pronto para ser manipulado. Todas as rotas para manipulação de dados pelo Postman estão disponibilizadas no arquivo **Projeto Certificadora.postman_collection**. Ele estará disponibilizado no repositório.