Atividade proposta – Construção de API

Nome: Victor Freitas Garces

**Introdução ao desenvolvimento de API**

A empresa “Casa da tecnologia”, que oferece produtos na área da tecnologia, deseja a construção de uma aplicação API REST interligada com banco de dados. Tal estrutura deve ser capaz de adicionar, listar, atualizar e deletar informações e dados dentro do banco de dados da instituição.

**Estrutura do projeto**

A estrutura do projeto tem como objetivo lista como a API está modelada, quais as pastas utilizadas e como se deu a construção da aplicação.

Atividade-proposta-desenvolvimentoAPI

|\_ config

|\_.env

|\_Controllers

|\_controllersRouters

|\_database

|\_inventory\_db

|\_.gitignore

|\_.env

\_\_Serve.js

\_\_Package

\_\_Package-lock

**Configurações de ambiente**

1. Variáveis de ambiente

As variáveis de ambiente são aquelas usadas para configuração do banco de dados, que permite a conexão e a interligação com o mesmo.

DB\_HOST=localhost

DB\_USERNAME=root

DB\_PASSWORD=07112001

DB\_NAME=inventory\_db

PORT=5000

1. Dependências

As dependências se referem a extensões e bibliotecas que foram utilizadas e são essenciais para a construção da aplicação que interliga as ações do usuário com movimentações fazendo uso das requisições HTTP. Desta forma teremos:

Npm init -y

Npm install cors body-parser dotenv express

Npm install nodemon –save--dev

**Banco de dados**

Na estruturação do banco de dados teremos apenas a tabela de “products”, na seguinte estrutura:

-- ATIVIDADE PROPOSTA - CONSTRUÇÃO DE API

create database inventory\_db;

USE inventory\_db;

-- criando a tabela products

CREATE TABLE products (

id\_product int(5) primary key auto\_increment,

name\_product varchar(255),

description\_product text,

category varchar(100),

price\_product decimal(5,2),

quant\_product int,

data\_validade date

);

-- inserindo produtos na tabela products

INSERT INTO products (name\_product, description\_product, category, price\_product, quant\_product, data\_validade) VALUES

('Notebook', 'Notebook Dell Inspiron 15 3000', 'Eletrônicos', 250.00, 10, '2025-12-31'),

('Smartphone', 'Smartphone Samsung Galaxy S21', 'Eletrônicos', 350.00, 20, '2025-12-31'),

('Geladeira', 'Geladeira Brastemp Frost Free', 'Eletrodomésticos', 450.00, 5, '2028-06-30'),

('Microondas', 'Microondas Electrolux 30L', 'Eletrodomésticos', 800.00, 15, '2027-08-15'),

('Cadeira Gamer', 'Cadeira Gamer DXRacer', 'Móveis', 150.00, 25, '2030-05-20'),

('Mesa de Escritório', 'Mesa de Escritório em L', 'Móveis', 700.00, 8, '2030-05-20'),

('Livro', 'Livro - Clean Code', 'Livros', 120.00, 50, '2050-01-01'),

('Caderno', 'Caderno Universitário 200 folhas', 'Papelaria', 20.00, 100, '2030-12-31'),

('Mouse', 'Mouse Gamer Logitech G502', 'Acessórios de Computador', 250.00, 30, '2029-11-11'),

('Teclado', 'Teclado Mecânico Razer', 'Acessórios de Computador', 500.00, 20, '2029-11-11');

**Configurações do projeto**

* **Configuração do banco de dados**

Na configuração do banco de dados, teremos o arquivo db.js sendo responsável por arquitetar e permitir a interligação entre o banco de dados (inventory\_db) com a API.

//INICIANDO CONFIGURAÇÃO DO BANCO DE DADOS

    //Importando bibliotecas

    const mysql2 = require('mysql2');

    //configurando conexão

    const db = mysql2.createConnection({

        host: process.env.DB\_HOST,

        user: process.env.DB\_USERNAME,

        password: process.env.DB\_PASSWORD,

        database: process.env.DB\_NAME

    }

    )

    db.connect ((err) => {

        if(err) {

            console.log('Erro ao conectar', err)

        return};

        console.log(`Conectado com sucesso ao banco de dados ${process.env.DB\_NAME}`)

    });

    //exportando a conexão do banco de dados

    module.exports=db;

* **Controlando as rotas(controllerRouter)**

**Teste da API**