UNIVERSIDADE SALVADOR – UNIFACS

SISTEMAS DISTRIBUÍDOS e MOBILE

IGOR DOS SANTOS

KEVIN SANTOS

MARISSA DE PAULA

NATHALIA DE LIMA

RAFAEL SANTANA

RELATÓRIO DO TRABALHO DA AVALIAÇÃO 3 – A3

SALVADOR

2023

IGOR DOS SANTOS - R.A: 12723213443

KEVIN SANTOS - R.A: 12722130284

MARISSA DE PAULA - R.A: 12722213256

NATHALIA DE LIMA - R.A: 12722210839

RAFAEL SANTANA - R.A: 12722213062

RELATÓRIO DO TRABALHO DA AVALIAÇÃO 3 – A3

Este relatório contém informações gerais da atividade proposta para entrega da Avaliação 3 da Unidade Curricular Sistemas Distribuídos e Mobile.

Professor: Adailton de Jesus Cerqueira Junior e Wellington Lacerda Silveira da Silva.

SALVADOR

2023

SUMÁRIO

INTRODUÇÃO

Neste relatório está presente informações utilizadas para o desenvolvimento da aplicação proposta pelos professores Adailton de Jesus e Wellington Lacerda na Unidade Curricular de Sistemas Distribuídos e Mobile. Foi proposta uma aplicação que simule a captação de dados e venda de uma rede de lojas. - A nossa escolha foi a venda de frutas e legumes por uma loja do tipo ‘HortiFruti’.

A linguagem utilizada fora JavaScript, e algumas das tecnologias utilizadas foram Docker, NodeJs, VsCode, PostgreSQL.

DESENVOLVIMENTO

Nesta avaliação, com a proposta de simulação sendo a captação de dados de venda de uma loja, o tema utilizado por esta equipe foi “HortiFruti”, com vendas de frutas e legumes.

A linguagem escolhida foi a JavaScript, pois foi essa a linguagem utilizada no decorrer do semestre atual. Então, a fim de minimizar conflitos gerados por confusões semânticas que diferentes linguagens podem causar com estudantes da área, os integrantes da equipe optaram por continuar a utilizar esta linguagem.

Sobre a arquitetura foi utilizado um banco de dados PostgreSQL, que é um dos bancos de dados relacionais de software livre mais utilizados. Este banco de dados roda em um container no Docker, que é uma plataforma de software que permite o desenvolvimento e gerenciamento de componentes de contêineres que combinam o código-fonte com as bibliotecas e estruturas do Sistema Operacional necessárias para a execução do código. Foi feita uma API REST para a integração das aplicações.

Neste trabalho, também foi utilizada interface Insominia para as requisições das APIs. O Node, que é um framework javascript, com uma das maiores comunidades de desenvolvedores e ampla utilização no mercado, também foi uma opção dos integrantes para o melhor funcionamento no desenvolvimento da aplicação proposta.

Utilizaremos as bibliotecas express, body-parser e cors. O express para criar as rotas e gerenciar as requisições e respostas da nossa aplicação e a body-parser que é um middleware para express, projetado para extrair o corpo das solicitações HTTP recebida pelo servidor e o cors para ajudar a lidar com cabeçalhos cors permitindo ou solici

INSTRUÇÕES PARA INSTALAÇÃO E EXECUÇÃO DA APLICAÇÃO

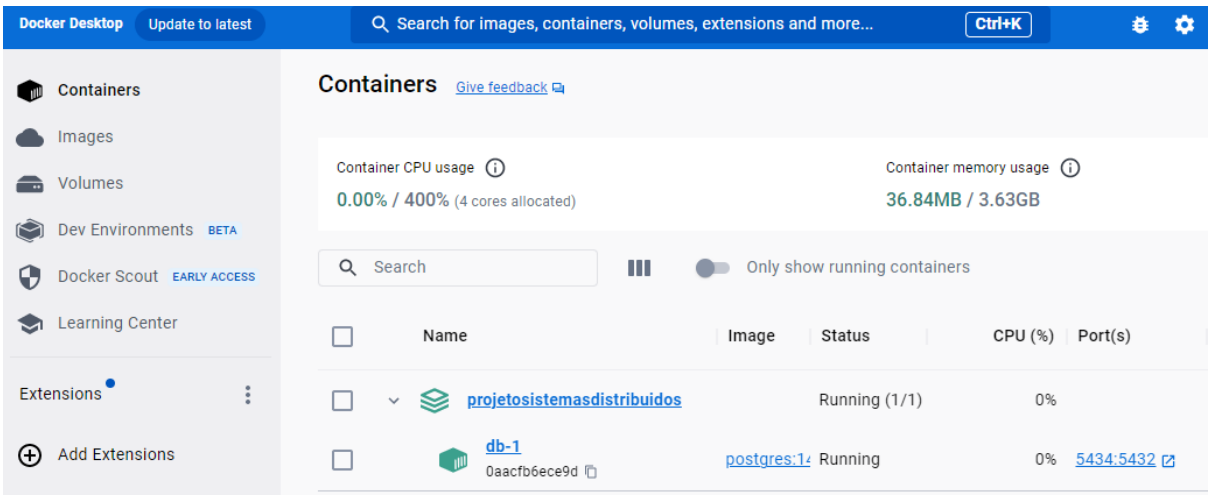
Para utilizar a aplicação é necessário seguir a seguinte instrução:

- Fazer o download do arquivo em ZIP disponibilizado no github.

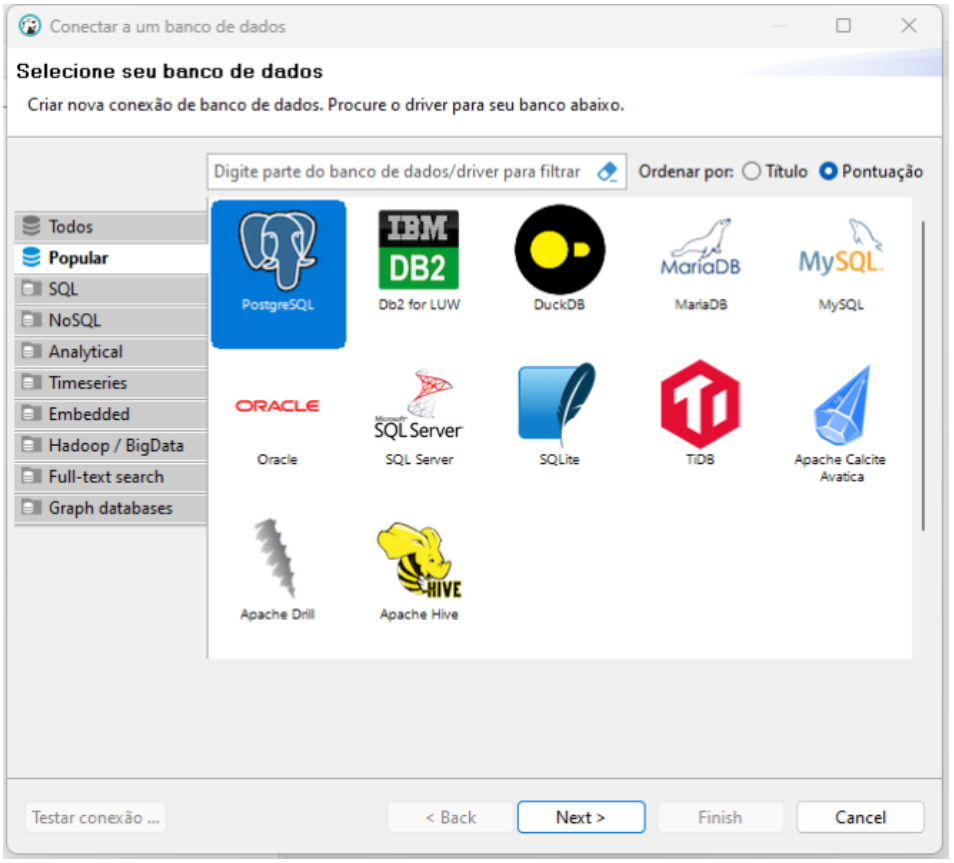
- Com o repositório baixado e aberto no vscode é necessário instalar as dependências, express, body-parser e cors utilizando o comando ‘npm install’ no terminal.

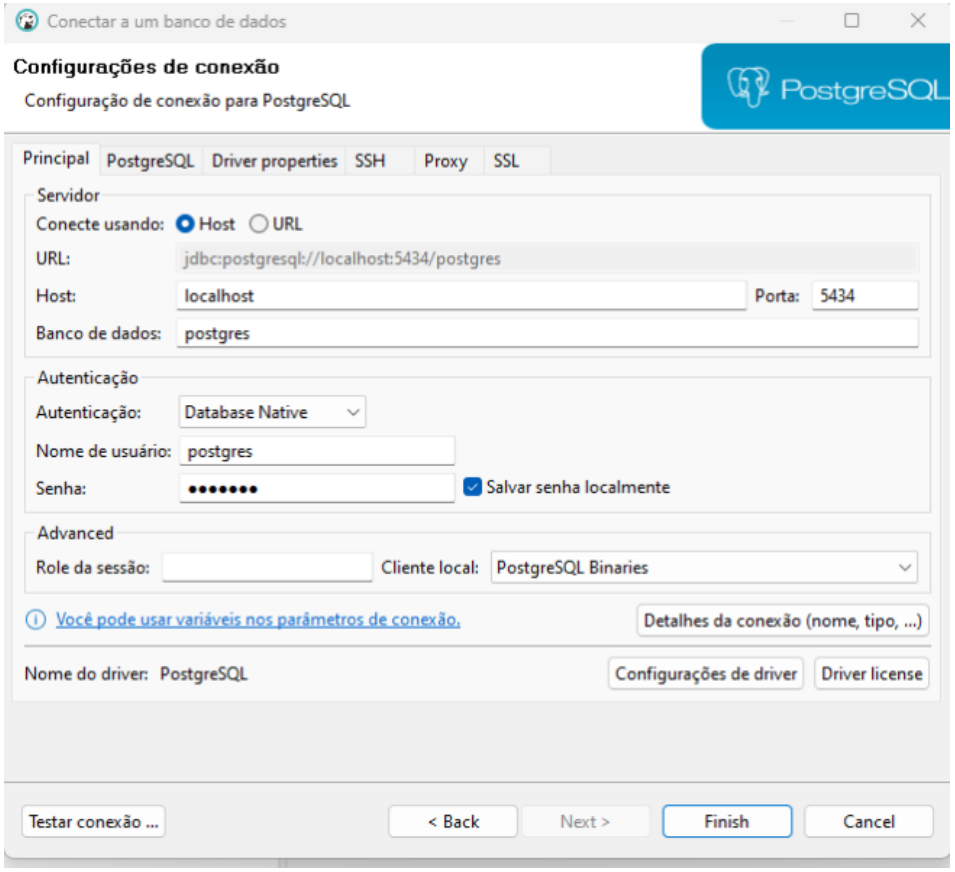
- No arquivo tem a pasta ‘docker-compose.yml’, dentro dele estão contidas as informações do Banco de Dados, como a senha e a porta - cujo onde o banco de dados está sendo rodado: porta 5434.

- Na pasta ‘package.json’ tem o script para iniciar o banco de dados. Para isso é necessário dar o seguinte comando no terminal: npm run iniciaBanco. Este comando irá fazer o download das imagens, criar e rodar o contêiner.



- Em um sistema de iniciador de banco de dados será necessário criar uma nova conexão do tipo ‘Postgresql’, porta ‘5434’, senha ‘example’ e clicar em ‘finish’.



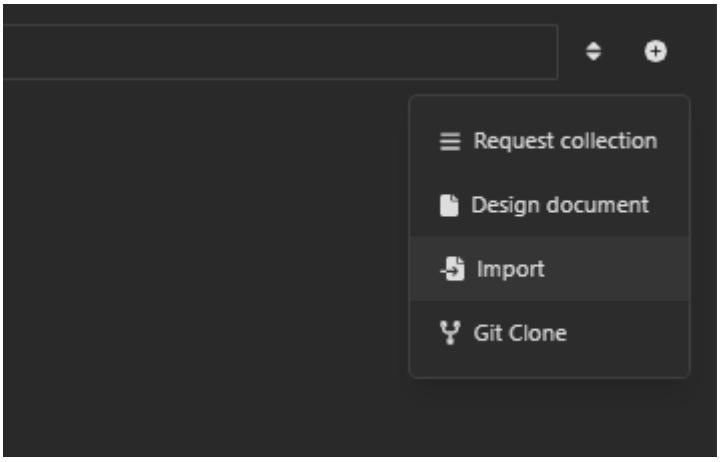


- Para criar tabelas: na pasta ‘src’, presente no repositório, tem uma pasta chamada ‘criaTabelas’. Para criar as tabelas de clientes, vendas, estoque e produtos é necessário dar o seguinte comando no terminal: npm criaTabela.

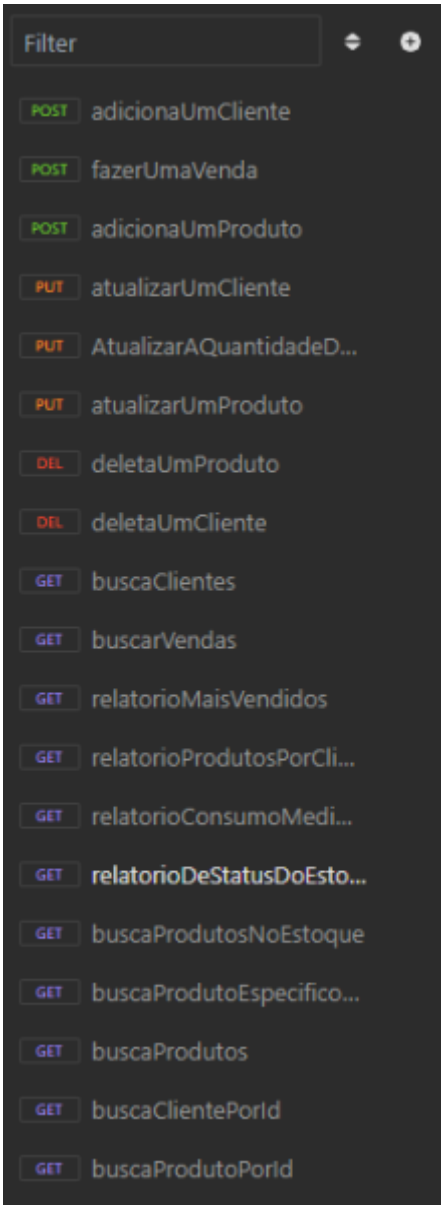
- No caminho postgres2 > postgres > schemas > public > tabelas, terá duas tabelas vazias: ‘clientes’ e ‘produtos’ - para preenchê-las deverá ir no vscode, no repositório e no terminal inserir o seguinte comando: npm run popularBanco.

- Na parte de interação com o sistema é necessário rodar o API, para isso é necessário inserir o seguinte comando no terminal: npm run dev. ele estará rodando na seguinte porta: http://localhost:3333/.

- Dentro do programa Insomnia deverá clicar no botão ‘+’, presente no canto superior direito e clicar na opção de importar o arquivo collection disponibilizado no GitHub na pasta auxiliar.



- Terá que abrir a collection ‘Sistemas Distribuídos’ e será possível verificar as rotas com os CRUDs, onde será possível buscar, criar, deletar e editar os clientes, produtos e estoque.



Clientes --

- adicionaUmcliente - rota que adiciona um cliente na tabela clientes.

- buscaClientes - rota que retorna todos os clientes salvos na tabela clientes

- buscaClientePorId - rota que retorna um cliente especifico da tabela clientes

- atualizarUmcliente - rota que atualiza os dados do cliente específico.

- deletaUmCliente - rota que apaga um cliente, mas se houver alguma venda associada ao mesmo, ele não será apagado pois geraria uma inconsistência no banco.

Produtos --

-adicionaUmProduto - rota que adiciona um produto na tabela produtos.

- buscaProdutos - rota que retorna todos os produtos salvos na tabela produtos.

- buscaProdutoPorId - rota que retorna um produto especifico da tabela produtos.

- atualizarUmProduto - rota que atualiza os dados do produto específico.

- deletaUmProduto- rota que apaga um produto, mas se houver alguma venda associada ao mesmo, ele não será apagado pois geraria uma inconsistência no banco.

Estoque --

- buscaProdutosNoEstoque - rota que busca os produtos no estoque. (Ao adicionar um produto, automaticamente é criado um registro na tabela de estoque, referenciando o produto com a quantidade inicial 0)

- buscaProdutoEspecificoNoEstoque - rota que busca um produto específico no estoque.

Vendas --

- fazerUmaVenda - rota que faz uma venda, apenas se houver a quantidade necessária do produto no estoque, após o registro da venda (compra) a quantidade do produto é subtraida pela quantidade vendida.

- relatorioMaisVendidos - retorna o relatório dos produtos mais vendidos

- relatorioProdutosPorcliente - retorna os dados das vendas relacionando produto com cliente.

- relatorioConsumoMedioCliente - retorna o consumo médio do cliente com base nas vendas (compras dele)

- relatorioDeStatusDoEstoque - retorna o status do produto no estoque, se for menor do que 2 = baixo, igual a 2= no limite, maior que 2 então está normal.

entes --

- adicionaUmcliente - rota que adiciona um cliente na tabela clientes.

- buscaClientes - rota que retorna todos os clientes salvos na tabela clientes

- buscaClientePorId - rota que retorna um cliente especifico da tabela clientes

- atualizarUmcliente - rota que atualiza os dados do cliente específico.

- deletaUmCliente - rota que apaga um cliente, mas se houver alguma venda associada ao mesmo, ele não será apagado pois geraria uma inconsistência no banco.

Produtos --

-adicionaUmProduto - rota que adiciona um produto na tabela produtos.

- buscaProdutos - rota que retorna todos os produtos salvos na tabela produtos.

- buscaProdutoPorId - rota que retorna um produto especifico da tabela produtos.

- atualizarUmProduto - rota que atualiza os dados do produto específico.

- deletaUmProduto- rota que apaga um produto, mas se houver alguma venda associada ao mesmo, ele não será apagado pois geraria uma inconsistência no banco.

Estoque --

- buscaProdutosNoEstoque - rota que busca os produtos no estoque. (Ao adicionar um produto, automaticamente é criado um registro na tabela de estoque, referenciando o produto com a quantidade inicial 0)

- buscaProdutoEspecificoNoEstoque - rota que busca um produto específico no estoque.

Vendas --

- fazerUmaVenda - rota que faz uma venda, apenas se houver a quantidade necessária do produto no estoque, após o registro da venda (compra) a quantidade do produto é subtraida pela quantidade vendida.

- relatorioMaisVendidos - retorna o relatório dos produtos mais vendidos

- relatorioProdutosPorcliente - retorna os dados das vendas relacionando produto com cliente.

- relatorioConsumoMedioCliente - retorna o consumo médio do cliente com base nas vendas (compras dele)

- relatorioDeStatusDoEstoque - retorna o status do produto no estoque, se for menor do que 2 = baixo, igual a 2= no limite, maior que 2 então está normal.

CONSIDERAÇÕES FINAIS

A conclusão bem-sucedida do projeto de desenvolvimento da aplicação "HortiFruti" em Sistemas Distribuídos e Mobile reforça a importância das escolhas estratégicas da equipe. A opção pela linguagem JavaScript, o uso do banco de dados PostgreSQL em conjunto com Docker, a implementação de uma API REST e a seleção de ferramentas como Insomnia e NodeJs destacam-se como decisões acertadas. Essas escolhas não apenas facilitaram o processo de desenvolvimento, mas também consolidaram a aplicação prática dos conceitos teóricos, evidenciando

a relevância de abordagens sólidas e tecnologias consolidadas em projetos dessa natureza.

BIBLIOGRAFIA

Título: What is PostgreSQL?

URL:https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-postgresql

Publicação: Microsoft Azure

Data de Consulta: 10 de dezembro de 2023

Título: Docker Overview

URL: https://www.ibm.com/br-pt/topics/docker

Publicação: IBM

Data de Consulta: 06 de dezembro de 2023

Título: Como criar uma API RESTful: Guia Completo

URL: https://www.hostinger.com.br/tutoriais/api-restful

Publicação: Hostinger

Data de Consulta: 06 de dezembro de 2023

Título: What is a REST API?

URL: https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api

Publicação: Red Hat

Data de Consulta: 13 de dezembro de 2023

Título: Usando o Insomnia para testar as requisições de nossas APIs

URL: https://www.hcode.com.br/blog/usando-insomnia-para-testar-as-requisicoes-de-nossas-apis

Publicação: HCode

Data de Consulta: 13 de dezembro de 2023