SERVICES TANQUES.JS FROMEND

const BASE\_URL = 'http://127.0.0.1:3000'

import { db } from '../services/firebase'

export const useTanques = () => {

const fetchTanque = async (id1) => {

  const options = {

    method: 'POST',

    headers: {

            'Content-type': 'application/json',

            'Authorization': `bearer ${localStorage.getItem("JWT\_TOKEN").replace(/['"]+/g, '')}`

    },

    body: JSON.stringify({id:id1}),

  }

  const response = await fetch(`${BASE\_URL}/devtanques`, options)

  console.log(response.content)

  return await response.json()

}

  const fetchTanques = async () => {

    const options = {

      method: 'GET',

      headers: {

        'Content-type': 'application/json',

      },

    }

    const response = await fetch(`${BASE\_URL}/tanques`,options)

    return await response.json()

  }

  const createTanques = async (form) => {

    const options = {

      method: 'POST',

      headers: {

        'Content-type': 'application/json',

         'Authorization': `bearer ${localStorage.getItem("JWT\_TOKEN").replace(/['"]+/g, '')}`

      },

      body: JSON.stringify({data:form})

    }

    const response = await fetch(`${BASE\_URL}/tanque`, options)

    return await response.json()

  }

   const removeTanques = async (id1) => {

    const options = {

      method: 'PUT',

      headers: {

        'Content-type': 'application/json',

         'Authorization': `bearer ${localStorage.getItem("JWT\_TOKEN").replace(/['"]+/g, '')}`

      },

      body: JSON.stringify({id:id1})

    }

    console.log(options)

    const response = await fetch(`${BASE\_URL}/deltanques`, options)

    return await response.json()

  }

 const editTanques = async (id,form) => {

  const options = {

    method: 'PUT',

    headers: {

      'Content-type': 'application/json',

       'Authorization': `bearer ${localStorage.getItem("JWT\_TOKEN").replace(/['"]+/g, '')}`

    },

    body: JSON.stringify({id:id,data:form})

  }

  console.log(options)

  const response = await fetch(`${BASE\_URL}/acttanques`, options)

  return await response.json()

}

  return {

    fetchTanques,

    createTanques,

    removeTanques,

    fetchTanque,

    editTanques

  }

}

BACKEND

TANQUES.CONTROLLER.JS

import { conexion } from "../conexion.js";

import { actulizartanqueSerializer,registrarTaqueSerializer } from "./serializers/tanques.serialize.js";

import {paginationSerializer} from "../util.js"

import { number } from "zod";

export const crearTanque = async (req, res) => {

const data = req.body.data

console.log(data);

data.capacidad = Number(req.body.data.capacidad)

data.HEIGHT\_PIES = Number(req.body.data.HEIGHT\_PIES)

data.metrica=Number(req.body.data.metrica)

console.log(data);

const dataValidada = registrarTaqueSerializer.parse(data);

const nuevoTanque = await conexion.tanques.create({data:dataValidada});

return res.json({

message:"Tanque creado exitosamente",

content: nuevoTanque,

})

}

export const devolverTanques = async (req,res) => {

const {page , perPage} = req.query

let skip,take

if(page && perPage){

//Cuantos elementos se debe saltar

skip = (Number(page) -1) \* Number(perPage)

//take cuantos elemetosse debe tomR LUEGO DEL SALTO

take = Number(perPage)

}

const filtros = {};

if (req.query.codigo){

filtros.codigo = {contains: req.query.codigo};

}

const totalTanques = await conexion.tanques.count({

where:filtros,

})

const ListaTanques = await conexion.tanques.findMany({

where:filtros,

skip,

take,

})

const pageInfo = paginationSerializer(

totalTanques,

Number(page),

Number(perPage)

)

return res.json({

message:"Lista de Tanques ",

content : ListaTanques,pageInfo,

})

}

export const borrartanque = async (req,res) => {

const tanqueborrado = await conexion.tanques.delete({

select: {

id :true,

codigo:true,

},

where: {

id:Number(req.body.id)

},

})

return res.json({

message:"Tanques borrado exitosamente",

content : tanqueborrado,

})

}

export const devolvertanque = async (req,res) => {

const taqueEncontrado = await conexion.tanques.findFirstOrThrow({

where: { id:Number(req.body.id)},

});

return res.json({

message:" Tanque encontrado exitosamente",

content : taqueEncontrado ,

}

)

}

export const actulizarTanque = async (req,res) => {

const data = req.body.data

data.capacidad = Number(req.body.data.capacidad)

data.HEIGHT\_PIES = Number(req.body.data.HEIGHT\_PIES)

data.metrica=Number(req.body.data.metrica)

const dataValidada =actulizartanqueSerializer.parse(data)

const tanqueActualizado = await conexion.tanques.update({

data : dataValidada,

where: {

id:Number(req.body.id)

},

})

return res.json({

message:"Tanque actualizado exitosamente",

content : tanqueActualizado,

})

}

SERILIZACION

import { TipoUsuario } from "@prisma/client"

import { z } from "zod"

export const registrarTaqueSerializer = z.object({

    codigo: z.string(),

    metrica: z.number(),

      capacidad: z.number(),

      HEIGHT\_PIES: z.number(),

  });

  export const actulizartanqueSerializer = z.object({

    metrica: z.number(),

    capacidad: z.number(),

    HEIGHT\_PIES: z.number(),

  })