favor realizar esto desde cero crear un mapa interactivo de Venezuela en nodejs usando Leaflet.js donde: El usuario pueda navegar de forma intuitiva, puede cambiar entre ciudades de venezuelcapitales al hacer clic en un botón, o seleccionar una capital de una lista entre los estados de Venezuela y sus capitales que son: Amazonas: Puerto Ayacucho, Barcelona, San Fernando de Apure, Maracay, Barinas, Ciudad Bolívar, Valencia, San Carlos, Tucupita, Coro, San Juan de los Morros, Barquisimeto, Mérida, Los Teques, Maturín, La Asunción, Guanare, Cumaná, San Cristóbal, Trujillo, La Guaira, San Felipe, Maracaibo; solo se vera el pais al inicio, y al seleccecionar una capital solo mostrara esa ciudad, queda la lista para selección , Incluir sistema de ping a una lista de clientes que estará guardada en un archivo .json con las variables de clientes que incluye por ejemplo: { "id": 1, "ciudad": "Valencia", "nombreCliente": "Valencia Ab 281893 Colgate Palmolive Michelena 100Mb ", "direccion": "10.29.156.102", "coordenadas": { "lat": 10.162, "lng": -68.007 } }, El resultado del pin se guardara en otro archivo llamado Pingnodos.json y se mostrando en el mapa la ubicación correspondiente en color verde si está conectado o color rojo si esta desconectado Reiniciando los ping y rescribiendo los archivos json cada 10 minutos asi se refrescara el estatus de resultados en el mapa una vez que se actualice el archivo Pingnodos.json

corregido la pequeña aplicación debe tener en formato .json, lista de clientes, lista de ciudades, lista de resultados los clientes deben tener id, nombre, sede, ip, coordenada, la lista de ciudad debe tener nombre de ciudad, nombre de estado y coordenada, la lista de resultado debe guardar el resultado de ping conectado o desconectado luego de hacer ping a la ip de la lista de clientes.

Una vez realizado los ping de la lista de clientes y guardada el resultado en la lista de resultados.json

Mostrara un mapa de Venezuela en colo rojo para resultado desconectado y color azul para resultado conectado… mostrara una lista de ciudades que se puede seleccionar manualmente con los datos de las ciudades en el archivo json

El mapa elegido debe cargar de la lista de resultados el resultado actual del ping

Y realizara la acción de ping cada 10 minutos despues de realizar el ping, actualizara los resultados en todos los mapa, favor cagara un registro de ejemplo en la lista de ciudad y de clientes para yo cargar el resto manualmente

Vamos a construir un mapa interactivo de Venezuela utilizando [Node.js](https://Node.js) y [Leaflet.js](https://Leaflet.js) desde cero. El objetivo será permitir la navegación intuitiva entre las capitales de los estados de Venezuela y mostrar un sistema de ping para clientes. Aquí tienes los pasos detallados:

**Paso 1: Configuración del proyecto**

1. **Crear un directorio del proyecto** y navegar a él:

sh

mkdir MapaVenezuela

cd MapaVenezuela

1. **Inicializar un proyecto** [**Node.js**](https://Node.js):

sh

npm init -y

1. **Instalar las dependencias necesarias**:

sh

npm install express axios

**Paso 2: Crear el servidor con Express**

Crea un archivo llamado server.js y configura un servidor básico:

javascript

// server.js

const express = require('express');

const axios = require('axios');

const fs = require('fs');

const app = express();

const port = 3000;

app.use(express.static('public'));

app.get('/ping', (req, res) => {

const clientes = JSON.parse(fs.readFileSync('clientes.json', 'utf8'));

const resultados = [];

clientes.forEach(cliente => {

axios.get(`http://${cliente.direccion}`)

.then(response => {

resultados.push({ ...cliente, estado: 'conectado' });

})

.catch(error => {

resultados.push({ ...cliente, estado: 'desconectado' });

});

});

setTimeout(() => {

fs.writeFileSync('Pingnodos.json', JSON.stringify(resultados, null, 2));

res.send(resultados);

}, 5000); // espera a que todos los pings terminen

});

app.listen(port, () => {

console.log(`Servidor corriendo en http://localhost:${port}`);

});

**Paso 3: Crear el archivo HTML y scripts**

Crea una carpeta llamada public y dentro de ella un archivo index.html:

html

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Mapa Interactivo de Venezuela</title>

<link rel="stylesheet" href="https://unpkg.com/leaflet/dist/leaflet.css" />

<style>

#map { height: 600px; width: 100%; }

.controls { margin-bottom: 10px; }

</style>

</head>

<body>

<h1>Mapa Interactivo de Venezuela</h1>

<div class="controls">

<select id="capitales"></select>

<button onclick="irCapitalAnterior()">Anterior</button>

<button onclick="irCapitalSiguiente()">Siguiente</button>

</div>

<div id="map"></div>

<script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>

<script src="app.js"></script>

</body>

</html>

**Paso 4: Crear el script de Leaflet**

Dentro de la carpeta public, crea un archivo llamado app.js:

javascript

// app.js

const map = L.map('map').setView([6.4238, -66.5897], 6);

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

maxZoom: 19,

}).addTo(map);

const estados = [

{ nombre: "Amazonas", capital: "Puerto Ayacucho", coordenadas: { lat: 5.6639, lng: -67.6236 } },

{ nombre: "Anzoátegui", capital: "Barcelona", coordenadas: { lat: 10.1333, lng: -64.6833 } },

{ nombre: "Apure", capital: "San Fernando de Apure", coordenadas: { lat: 7.8961, lng: -67.4724 } },

{ nombre: "Aragua", capital: "Maracay", coordenadas: { lat: 10.2469, lng: -67.5958 } },

{ nombre: "Barinas", capital: "Barinas", coordenadas: { lat: 8.6226, lng: -70.2075 } },

{ nombre: "Bolívar", capital: "Ciudad Bolívar", coordenadas: { lat: 8.1292, lng: -63.5409 } },

{ nombre: "Carabobo", capital: "Valencia", coordenadas: { lat: 10.162, lng: -68.007 } },

{ nombre: "Cojedes", capital: "San Carlos", coordenadas: { lat: 9.6625, lng: -68.5958 } },

{ nombre: "Delta Amacuro", capital: "Tucupita", coordenadas: { lat: 9.0622, lng: -62.0529 } },

{ nombre: "Falcón", capital: "Coro", coordenadas: { lat: 11.4069, lng: -69.6737 } },

{ nombre: "Guárico", capital: "San Juan de los Morros", coordenadas: { lat: 9.9117, lng: -67.3581 } },

{ nombre: "Lara", capital: "Barquisimeto", coordenadas: { lat: 10.0739, lng: -69.3228 } },

{ nombre: "Mérida", capital: "Mérida", coordenadas: { lat: 8.5984, lng: -71.1500 } },

{ nombre: "Miranda", capital: "Los Teques", coordenadas: { lat: 10.3442, lng: -67.0437 } },

{ nombre: "Monagas", capital: "Maturín", coordenadas: { lat: 9.7457, lng: -63.1832 } },

{ nombre: "Nueva Esparta", capital: "La Asunción", coordenadas: { lat: 11.0333, lng: -63.8628 } },

{ nombre: "Portuguesa", capital: "Guanare", coordenadas: { lat: 9.0429, lng: -69.7421 } },

{ nombre: "Sucre", capital: "Cumaná", coordenadas: { lat: 10.4542, lng: -64.1825 } },

{ nombre: "Táchira", capital: "San Cristóbal", coordenadas: { lat: 7.7669, lng: -72.2250 } },

{ nombre: "Trujillo", capital: "Trujillo", coordenadas: { lat: 9.3669, lng: -70.4361 } },

{ nombre: "Vargas", capital: "La Guaira", coordenadas: { lat: 10.5969, lng: -66.9372 } },

{ nombre: "Yaracuy", capital: "San Felipe", coordenadas: { lat: 10.3406, lng: -68.7425 } },

{ nombre: "Zulia", capital: "Maracaibo", coordenadas: { lat: 10.6316, lng: -71.6406 } }

];

let currentCapitalIndex = 0;

function irACapital(index) {

const estado = estados[index];

map.setView([estado.coordenadas.lat, estado.coordenadas.lng], 10);

}

function irCapitalAnterior() {

currentCapitalIndex = (currentCapitalIndex - 1 + estados.length) % estados.length;

irACapital(currentCapitalIndex);

}

function irCapitalSiguiente() {

currentCapitalIndex = (currentCapitalIndex + 1) % estados.length;

irACapital(currentCapitalIndex);

}

const selectCapitales = document.getElementById('capitales');

estados.forEach((estado, index) => {

const option = document.createElement('option');

option.value = index;

option.textContent = `${estado.capital}`;

selectCapitales.appendChild(option);

});

selectCapitales.addEventListener('change', (e) => {

currentCapitalIndex = parseInt(e.target.value);

irACapital(currentCapitalIndex);

});

estados.forEach(estado => {

const marker = L.marker([estado.coordenadas.lat, estado.coordenadas.lng])

.addTo(map)

.bindPopup(`${estado.capital}`);

marker.on('mouseover', () => marker.openPopup());

marker.on('mouseout', () => marker.closePopup());

});

function actualizarMarcadores() {

fetch('/ping')

.then(response => response.json())

.then(data => {

data.forEach(cliente => {

const color = cliente.estado === 'conectado' ? 'green' : 'red';

const marcador = L.circleMarker([cliente.coordenadas.lat, cliente.coordenadas.lng], { color: color }).addTo(map);

marcador.bindPopup(`${cliente.nombreCliente}`);

});

});

}

// Actualizar marcadores cada 10 minutos

setInterval