



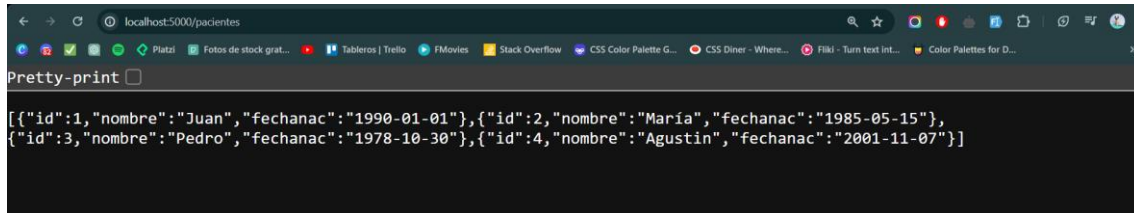
Apirest Pacientes en NodeJs

Desarrollar el ejercicio para crear 3 objetos de un paciente, utilizando las clases ServicePaciente y Paciente, consumir estos datos utilizando apiREST desde NodeJs

Enlace de GitHub: <https://github.com/JuanApazaG/Ing-de-software0.2.git>

Resultado que nos sale a

Cuando entramos al servidor <http://localhost:5000/pacientes> podemos observar que nos devuelve el resultado



Este es el código que implementamos de ecampus y agregando cosas de otros laboratorios anteriores.

```
app.js LAB9 U Paciente.js U PacienteService.js U app.js .\ M X package.json
app.js > ...
1  /*
2  Lab 009: Ingenieria de Software
3  ApiREST Pacientes en NodeJs
4  @autor Juan Agustin Apaza Guzman
5  @version 0.1
6  @date 26/04/2024
7  */
8  const express = require("express");
9  const app = express();
10
11  app.listen(5000, () => {
12    console.log("Servidor escuchando en http://localhost:5000");
13  });
14
15  app.get("/", (req, res) => {
16    res.send("Hola mundo \n Juan Agustin Apaza");
17  });
18
19  // Importar PacienteService usando la ruta relativa
20  const PacienteService = require('./PacienteService');
21
22  const pacienteService = new PacienteService();
23
24  app.get('/pacientes', (req, res) => {
25    const pacientes = pacienteService.getAllPacientes();
26    res.json(pacientes);
27  });
28
29  app.get('/pacientes/:id', (req, res) => {
30    const id = parseInt(req.params.id);
31    const paciente = pacienteService.getPacienteById(id);
32    if (paciente) {
33      res.json(paciente);
34    } else {
35      res.status(404).send('Paciente no encontrado');
36    }
37  });
38
39  const PORT = process.env.PORT || 3000;
40  app.listen(PORT, () => {
41    console.log(`Servidor en ejecución en el puerto ${PORT}`);
42  });
43
PROBLEMAS 1 SALIDA TERMINAL PUERTOS COMENTARIOS CONSOLA DE DEPURACIÓN
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)
at node:internal/main/run_main_module:23:47
Node.js v18.15.0
PS E:\Juana\Desktop\Ing de Sistemas\7mo Semestre\Ing de Software\Laboratorios JUAN> node app.js
Servidor escuchando en http://localhost:5000
Servidor en ejecución en el puerto 3000
[]
Live Share Java: Ready GIST [Create Profile]
```



Lo que sigue es lo que el ing también nos brinda

```
app.js LAB9 U  JS Paciente.js U X JS PacienteService.js U
JS Paciente.js > Paciente > constructor
1 class Paciente {
2   constructor(id, nombre, fechanac) {
3     this.id = id;
4     this.nombre = nombre;
5     this.fechanac = fechanac;
6   }
7 }
8
9
10 module.exports = Paciente;
11
```

```
app.js LAB9 U  JS Paciente.js U  JS PacienteService.js U X  JS app.js .\ M  package
JS PacienteService.js > PacienteService > constructor
1 const Paciente = require('./Paciente.js');
2
3 class PacienteService {
4   constructor() {
5     this.pacientes = [
6       new Paciente(1, 'Juan', '1990-01-01'),
7       new Paciente(2, 'María', '1985-05-15'),
8       new Paciente(3, 'Pedro', '1978-10-30'),
9       new Paciente(4, 'Agustin', '2001-11-07')
10    ];
11  }
12
13   getAllPacientes() {
14     return this.pacientes;
15   }
16
17   getPacienteById(id) {
18     return this.pacientes.find(paciente => paciente.id === id);
19   }
20 }
21
22 module.exports = PacienteService;
23
```