

Manual De Tecnico

28/10/2023

Samuel Nehemias Coyoy Perez

202200198

Contenido

Objetivos	3
Específicos	3
Generales	3
Especificación Técnica	3
Requisitos Hardware	3
Requisitos Software	3
LOGICA PARA REALIZACION DEL PROGRAMA	4
Clases Utilizadas	4
Librerías Utilizadas	10

Objetivos

Específicos

- Gestión de usuarios

Generales

- Gestion de citas citas

- Modificacion de usuarios

- Creación de reportes

Especificación Tecnica

Requisitos Hardware

- Mouse

- Teclado

- Monitor

Requisitos Software

- Sistema Operativo: Windows 10

- Herramientas: IDE NetBeans

- Lenguaje de Programacion: JAVA

LOGICA PARA REALIZACION DEL PROGRAMA

Clases Utilizadas

Auth.Routes

```
const express = require('express');
const AppData = require('../AppData');

const router = express.Router();

//Logear Usuario
router.post('/login', (req, res) => {
  const userName = req.body.uNombre;
  const password = req.body.uPassword;

  for (usuario of AppData.usuarios) {
    if (usuario.userName === userName && usuario.password === password) {
      return res.status(200).json({ usuario});
    }
  }

  return res.status(400).json({
    error: 'Datos incorrectos.'
  })
})

//Registro Usuario
```

```

router.post('/registro', (req, res) => {
  const { v4: uuidv4 } = require('uuid');
  const id = uuidv4();
  const nombre = req.body.uNombreR;
  const apellido = req.body.uApellido;
  const userName = req.body.uUserName;
  const fechaNacimiento = req.body.uFechaNacimiento;
  const genero = req.body.uGenero;
  const password = req.body.uPassword;
  const telefono = req.body.uTelefono;
  const tipoUsuario = "PACIENTE";

  const usuarioExistente = AppData.usuarios.find((usuario) => usuario.userName
=== userName);

  if (usuarioExistente) {
    return res.status(400).json({
      error: 'Usuario ya registrado.'
    });
  } else {

    if (password.length > 7) {
      const nuevoUsuario = {
        id,
        nombre,
        apellido,
        userName,
        fechaNacimiento,
        genero,

```

```

        password,
        telefono,
        tipoUsuario
    }
    AppData.usuarios.push(nuevoUsuario);
    return res.status(200).json(nuevoUsuario);
} else {
    return res.status(400).json({
        error: 'La contraseña debe tener al menos 8 caracteres.'
    });
}

}

})

```

```

module.exports = router;

```

Pacientes.routes

```

const express = require('express');
const AppData = require('../AppData');
const router = express.Router();

//Modificar Usuario
router.post('/modificarPerfil', (req, res) => {
    const nombre = req.body.uNombreR;
    const apellido = req.body.uApellido;

```

```
const usurActual = req.body.UserActual;
const userName = req.body.uUserName;
const fechaNacimiento = req.body.uFechaNacimiento;
const password = req.body.uPassword;

const usuarioExistente = AppData.usuarios.find((usuario) => usuario.userName
=== userName);
if (usuarioExistente) {
  return res.status(400).json({
    error: 'Usuario ya registrado.'
  });
} else {
  for (usuario of AppData.usuarios) {
    if (usuario.userName === usurActual) {
      if (password.length > 7) {
        usuario.nombre = nombre;
        usuario.apellido = apellido;
        usuario.userName = userName;
        usuario.fechaNacimiento = fechaNacimiento;
        usuario.password = password;

        return res.status(200).json({ usuario });
      }
    }
  }
}
```

```

    }
  }

})

//Solicitar Cita
router.post('/citas', (req, res) => {
  const { v4: uuidv4 } = require('uuid');
  const id = uuidv4();
  const fecha = req.body.fecha;
  const hora = req.body.hora;
  const motivo = req.body.motivo;
  const estado = req.body.estado;

  //AQUI SE QUEDA VACIO MIENTRAS QUE ENFERMERA NO ACEPTA LA
  CITA

  const doctor = req.body.doctor;
  const idPaciente = req.body.idPaciente;

  const cita = {
    id,
    fecha,
    hora,
    motivo,
    estado,
    doctor,
    idPaciente
  }

  AppData.citas.push(cita);
  return res.status(200).json({ citas: AppData.citas });
});

```



```
})
```

```
router.post('/citasPaciente', (req, res) => {
```

```
    const citas = AppData.citas.filter(cita => {  
        return cita.idPaciente === req.body.paciente;  
    });
```

```
    return res.status(200).json({citas: AppData.citas });
```

```
})
```

```
// Ruta para obtener la lista de citas
```

```
router.get('/obtenerCitas', (req, res) => {
```

```
    return res.status(200).json({ citas: AppData.citas });  
});
```

```
router.get('/stockMedicinas', (req, res) => {
```

```
    const medicinas = AppData.medicinas.filter(medicina => {  
        return medicina.cantDisponible > 0;  
    });
```

```
    return res.status(200).json({ medicinas });
```

```
})
```

```
module.exports = router;
```

Librerías Utilizadas

Uuidv4

npx

