

GA7-220501096-AA5-EV01  
Diseño y Desarrollo de servicios web

Repositorio: <https://github.com/delahoz174/Apis-sena.git>

Evidencias:

- Registro de usuario:

```
// Ruta de registro
router.post('/register', async (req, res) => {
  try {
    const { username, password } = req.body;

    // Verificar si ya existe el usuario
    const userExist = await User.findOne({ username });
    if (userExist) {
      return res.status(400).json({ message: 'Usuario ya existe' });
    }

    // Encriptar la contraseña
    const hashedPassword = await bcrypt.hash(password, 10);

    // Crear usuario
    const newUser = new User({ username, password: hashedPassword });
    await newUser.save();

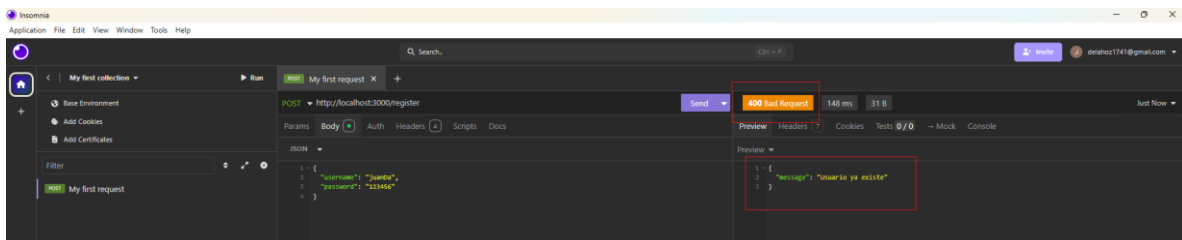
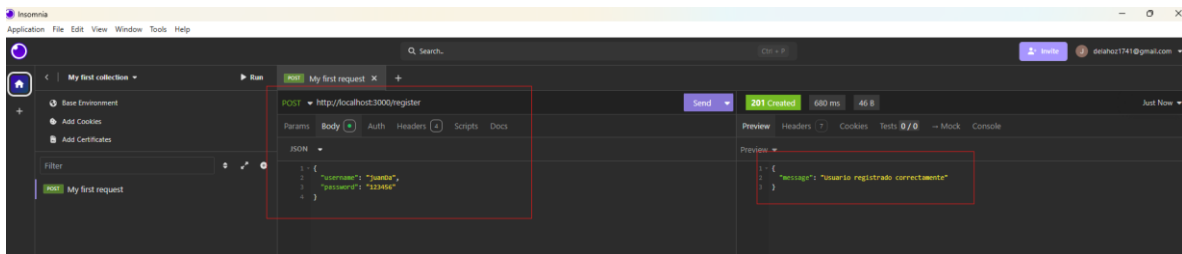
    res.status(201).json({ message: 'Usuario registrado correctamente' });
  } catch (error) {
    res.status(500).json({ message: 'Error en el servidor', error });
  }
});
```

- Consumo:

POST: <http://localhost:3000/register>

Body:

```
{
  "username": "juanDa",
  "password": "123456"
}
```



- Login

```
31 // Ruta de login
32 router.post('/login', async (req, res) => {
33   try {
34     const { username, password } = req.body;
35
36     // Buscar usuario
37     const user = await User.findOne({ username });
38     if (!user) {
39       return res.status(400).json({ message: 'Usuario no encontrado' });
40     }
41
42     // Comparar contraseñas
43     const isMatch = await bcrypt.compare(password, user.password);
44     if (!isMatch) {
45       return res.status(400).json({ message: 'Contraseña incorrecta' });
46     }
47
48     res.json({ message: 'Autenticación satisfactoria' });
49   } catch (error) {
50     res.status(500).json({ message: 'Error en el servidor', error });
51   }
52 });
53
54 module.exports = router;
55
```

- Consumo:

POST: <http://localhost:3000/login>

Body:

```
{  
  "username": "juanDa",  
  "password": "123456"  
}
```

