

# Codificación

---

## Botón

```
const request = require('supertest');
const assert = require('assert');
const express = require('express');
const { check, validationResult } = require('express-validator');

const app = express();
app.use(express.json());

const validaUsuario = () => {
  return [
    check('nombre').isLength({ min: 3, max: 14 }),
    check('apellido').isLength({ min: 10, max: 14 }),
    check('edad').isNumeric().withMessage("La edad debe ser numérica"),
    check('correo').isEmail().withMessage("El correo no tiene un formato válido")
  ];
};

app.post("/alumno", validaUsuario(), (req, res) => {
  const result = validationResult(req);
  if (result.isEmpty()) {
    res.json({ mensaje: "Respuesta a petición POST" });
  } else {
    res.json(result);
  }
});

// Prueba con Supertest
const userDataValid = {
  nombre: 'NombreValido',
  apellido: 'ApellidoValido',
  edad: 25,
  correo: 'correo@ejemplo.com'
};

const userDataInvalid = {
  nombre: 'N',
  apellido: 'ApellidoInvalido',
  edad: 'no-es-numeric',
  correo: 'correo-invalido'
};

request(app)
  .post('/alumno')
  .send(userDataValid)
  .expect(200)
```

```
.end((err, res) => {
  if (err) {
    console.error('Error en la prueba de datos válidos:', err);
  } else {
    assert.deepStrictEqual(res.body, { mensaje: "Respuesta a petición
POST" });
    console.log('Prueba de datos válidos exitosa');
  }
});

request(app)
  .post('/alumno')
  .send(userDataInvalid)
  .expect(200)
  .end((err, res) => {
    if (err) {
      console.error('Error en la prueba de datos inválidos:', err);
    } else {
      assert.deepStrictEqual(res.body.errors.length, 4);
      console.log('Prueba de datos inválidos exitosa');
    }
  });

app.listen(8084, () => {
  console.log("Servidor express escuchando en el puerto 8084");
});
```