



# ESPECIFICACIONES TÉCNICAS

## TEST APIS SPRINT 2

<b>Sprint - Deadline</b>	<b>2 - 3/11/23</b>
<b>Jira asociado</b>	<b>SCRUM-42</b>
<b>Responsable</b>	<b>Javier Huebra</b>
<b>Equipo</b>	<ul style="list-style-type: none"><li>● Miguel Ángel Maisares</li><li>● Lucas Sarfati</li><li>● Javier Huebra</li><li>● Matías Stewart Usher</li><li>● Facundo Oliva</li><li>● Koba Chajchir</li><li>● Yamil Leotta</li></ul>
<b>Versión</b>	1.0

## Contexto

Se efectúa un testeo mediante postman para constatar que los códigos de respuesta de los endpoints sean consecuentes con los resultados esperados en caso de respuesta favorable.

## Objetivo

Validar que las apis se están comportando de la manera esperada por el sistema.

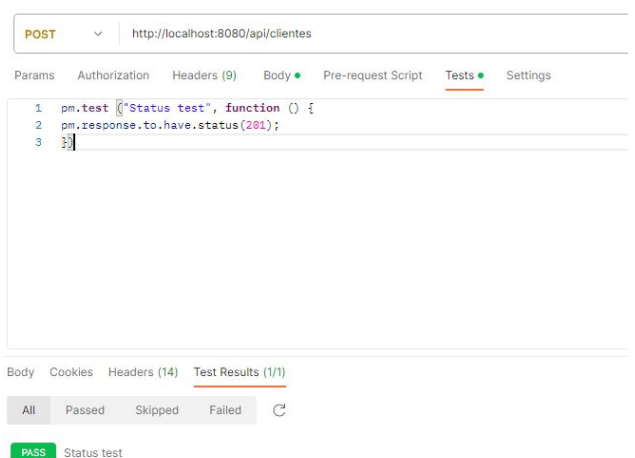
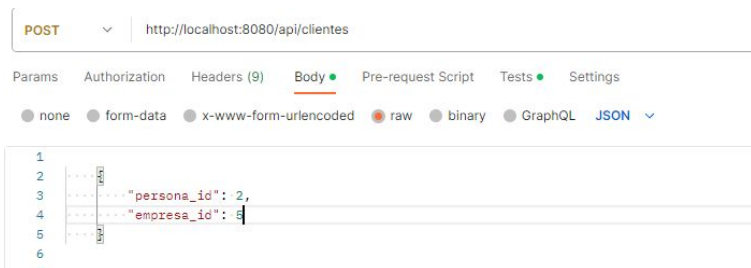
## Alcance

- Efectuar un script de testeo en la herramienta POSTMAN para:
- Clientes
- Personas
- Empresas
- Productos/Servicios
- Impuestos

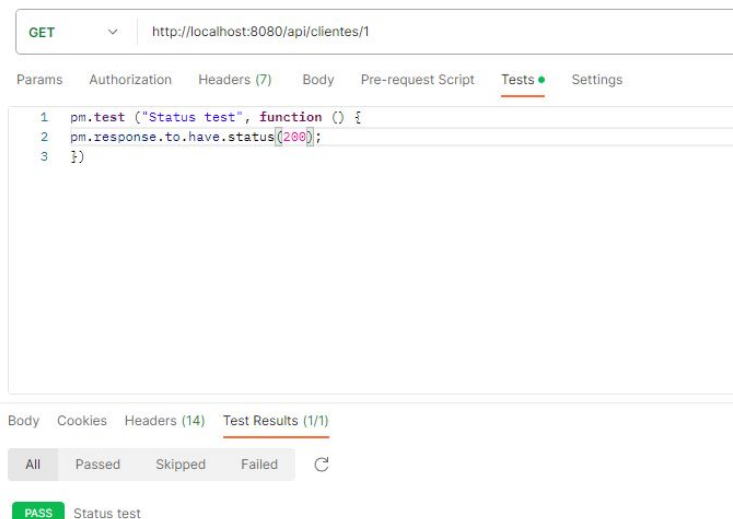
## Entregable

### Endpoint clientes: /api/clientes

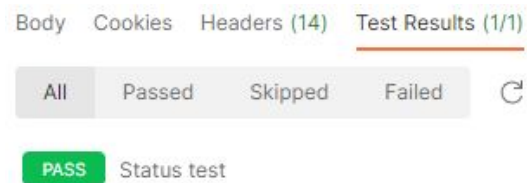
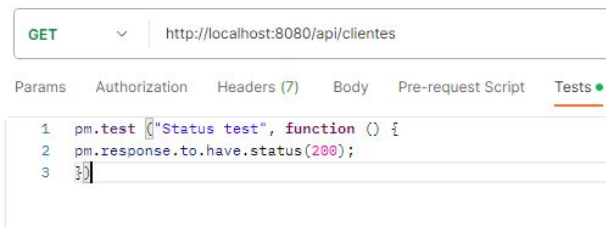
- **Creación de clientes:**
  - POST /api/clientes



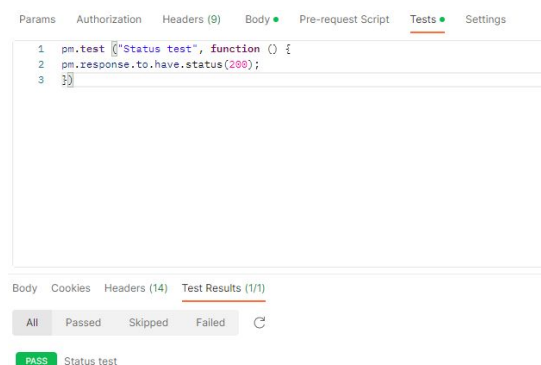
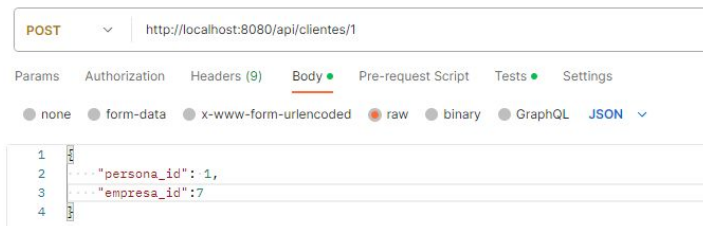
- **Seleccionar un cliente:**
  - GET /api/clientes/{id}



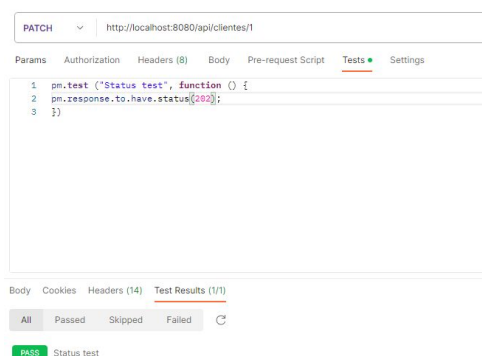
- **Listar todos los clientes:**
  - GET /api/clientes



- **Update cliente:**
  - POST /api/clientes/{id}



- **Delete cliente:**
  - PATCH /api/clientes/{id}



## Endpoint personas: /api/personas

- **Creación de personas:**
  - POST /api/personas

The screenshot shows a Postman interface for a POST request to `http://localhost:8080/api/personas`. The request body is a JSON object: `{ "nombre": "Persona", "apellido": "Ingresada", "dni": "41934987" }`. The test script contains: 

```
1 pm.test("Status test", function () {
2   pm.response.to.have.status(201);
3 })
```

 The Test Results tab shows a single test "Status test" that has passed.

- **Listar personas:**
  - GET /api/personas

The screenshot shows a Postman interface for a GET request to `http://localhost:8080/api/personas`. The test script contains: 

```
1 pm.test("Status test", function () {
2   pm.response.to.have.status(200);
3 })
```

 The Test Results tab shows a single test "Status test" that has passed.

- **Eliminar personas:**
  - PATCH /api/personas/{id}

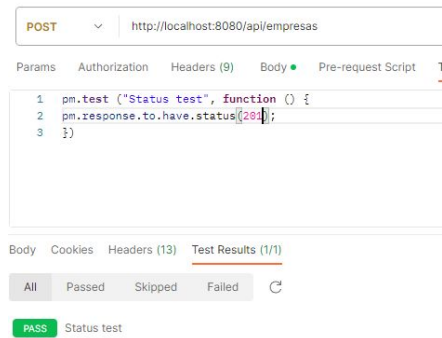
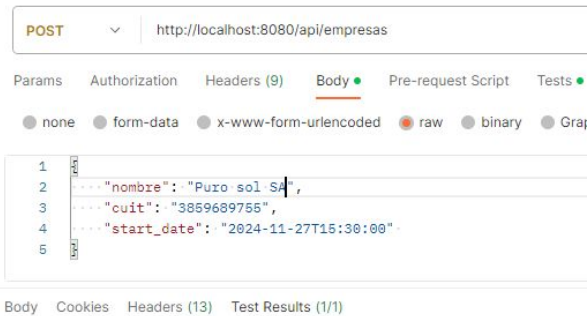
The screenshot shows a Postman interface for a PATCH request to `http://localhost:8080/api/personas/2`. The test script contains: 

```
1 pm.test("Status test", function () {
2   pm.response.to.have.status(202);
3 })
```

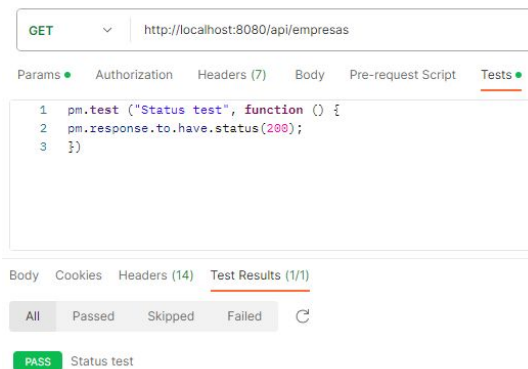
 The Test Results tab shows a single test "Status test" that has passed.

## Endpoint empresas: /api/empresas

- **Creación de empresa:**
  - POST /api/empresa



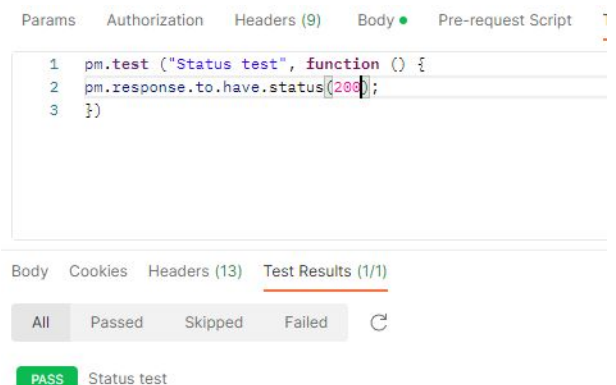
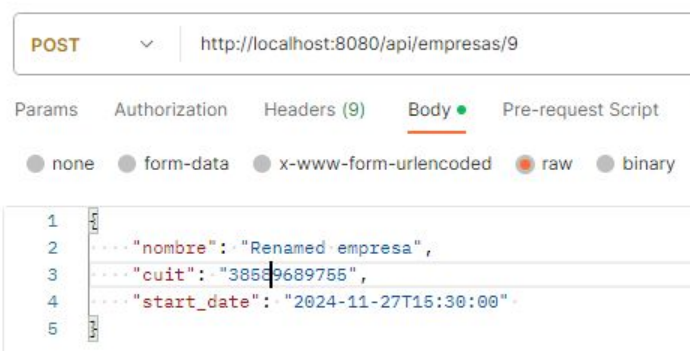
- **Listar empresas:**
  - GET /api/empresas



- **Eliminar empresas:**
  - PATCH /api/empresas/{id}

PENDIENTE

- **Update empresa:**
  - POST /api/empresas/{id}



## Endpoint Productos/Servicios: /api/prods\_servs

- **Crear producto/servicio:**
  - POST /api/prods\_servs

The screenshot shows a Postman interface for a POST request to `http://localhost:8080/api/prods_servs`. The request body is a JSON object: `{ "nombre": "Producto Nuevo", "costo": 125, "tipo": "PRODUCTO" }`. The 'Body' tab is selected, and the 'raw' format is chosen. To the right, the 'Tests' tab shows a status test: `pm.test(["Status test", function () { pm.response.to.have.status(201); }])`. The test results show 'PASS' for the 'Status test'.

- **Listar productos/servicios:**
  - GET /api/prods\_servs

The screenshot shows a Postman interface for a GET request to `http://localhost:8080/api/prods_servs`. The 'Tests' tab is selected, showing a status test: `pm.test("Status test", function () { pm.response.to.have.status(200); })`. The test results show 'PASS' for the 'Status test'.

- **Update productos/servicios:**
  - POST /api/prods\_servs/{id}

The screenshot shows a Postman interface for a POST request to `http://localhost:8080/api/prods_servs/1`. The request body is a JSON object: `{ "nombre": "Producto 1 Mod", "costo": 50 }`. The 'Body' tab is selected, and the 'raw' format is chosen.

- **Delete productos/servicios:**
  - PATCH /api/prods\_servs/{id}

The screenshot shows a Postman interface for a PATCH request to `http://localhost:8080/api/prods_servs/2`. The 'Tests' tab is selected, showing a status test: `pm.test(["Status test", function () { pm.response.to.have.status(200); }])`. The test results show 'PASS' for the 'Status test'.

## Endpoint Impuestos: /api/impuestos

- **Crear impuesto:**
  - POST /api/impuestos

POST ▼ http://localhost:8080/api/impuestos

Params Authorization Headers (9) **Body** ● Pre-request Script

● none ● form-data ● x-www-form-urlencoded ● raw ● binary

```
1 {
2   "nombre": "Impuesto nuevo",
3   "porcentaje": 20
4 }
```

Params Authorization Headers (9) **Body** ● Pre-request Script **Tests** ●

```
1 pm.test ("Status test", function () {
2   pm.response.to.have.status(201);
3 })
```

Body Cookies Headers (13) **Test Results (1/1)**

All Passed Skipped Failed ↻

**PASS** Status test

- **Listar impuestos:**
  - GET /api/impuestos

GET ▼ http://localhost:8080/api/impuestos

Params Authorization Headers (7) **Body** Pre-request

```
1 pm.test ("Status test", function () {
2   pm.response.to.have.status(200);
3 })
```

Body Cookies Headers (14) **Test Results (1/1)**

All Passed Skipped Failed ↻

**PASS** Status test

- **Update impuesto:**
  - POST /api/impuestos/{id}

POST ▼ http://localhost:8080/api/impuestos/2

Params Authorization Headers (9) **Body** ● Pre-request

● none ● form-data ● x-www-form-urlencoded ● raw ●

```
1 {
2   "nombre": "iva editado",
3   "porcentaje": 15
4 }
```

Params Authorization Headers (9) **Body** ● Pre-request Script **Tests** ●

```
1 pm.test ("Status test", function () {
2   pm.response.to.have.status(200);
3 })
```

Body Cookies Headers (13) **Test Results (1/1)**

All Passed Skipped Failed ↻

**PASS** Status test

- **Eliminar impuesto:**
  - PATCH /api/impuestos/{id}

PATCH ▼ http://localhost:8080/api/impuestos/2

Params Authorization Headers (8) **Body** Pre-request Script **Tests** ●

```
1 pm.test ("Status test", function () {
2   pm.response.to.have.status(202);
3 })
```

Body Cookies Headers (13) **Test Results (1/1)**

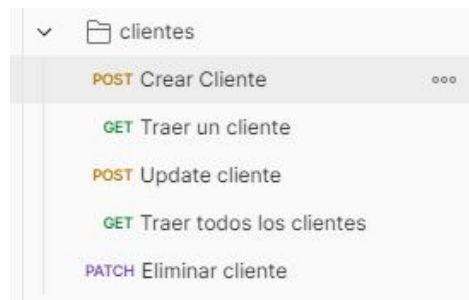
All Passed Skipped Failed ↻

**PASS** Status test



## Puntos de control

- Se validaron los endpoints de clientes en postman:



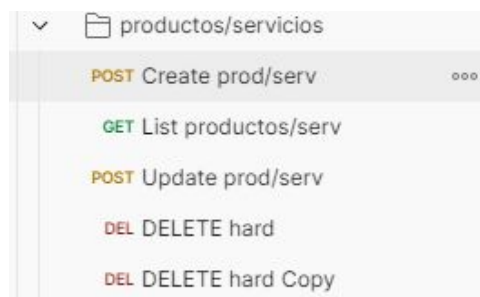
- Se validaron los endpoints de personas en postman:



- Se validaron los endpoints de empresas en postman:



- Se validaron los endpoints de productos/servicios en postman:



- Se validaron los endpoints de impuestos en postman:

