



## Prueba de Automatización

### 1. Prueba SQL.

**Enunciado:** Sean las siguientes tablas de una base de datos

**Empleado** (nDIEmp, nomEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

**Departamento** (codDepto, nombreDpto, ciudad, director)

codDepto	nombreDpto	ciudad	director
1000	GERENCIA	CALI	31.840.269
1500	PRODUCCIÓN	CALI	16.211.383
2000	VENTAS	CALI	31.178.144
3000	INVESTIGACIÓN	CALI	16.759.060
3500	MERCADEO	CALI	22.222.222
2100	VENTAS	POPAYAN	31.751.219
2200	VENTAS	BUGA	768.782
2300	VENTAS	CARTAGO	737.689
4000	MANTENIMIENTO	CALI	333.333.333
4100	MANTENIMIENTO	POPAYAN	888.888
4200	MANTENIMIENTO	BUGA	11.111.111
4300	MANTENIMIENTO	CARTAGO	444.444

nDIEmp	nomEmp	sexEmp	fecNac	fecIncorporacion	salEmp	comisionE	cargoE	jefeID	codDepto
31.840.269	'María Rojas'	'F'	15/01/59	16/05/1990	6250000	1500000	'Gerente'	NULL	'1000'
16.211.383	'Luis Pérez'	'M'	25/02/56	01/01/2000	5050000	0	'Director'	31.840.269	'1500'
31.178.144	'Rosa Angulo'	'F'	15/03/57	16/08/1998	3250000	3500000	'Jefe Ventas'	31.840.269	'2000'
16.759.060	'Dario Casas'	'M'	05/04/60	01/11/1992	4500000	500000	'Investigador'	31.840.269	'3000'
22.222.222	'Carla López'	'F'	11/05/75	16/07/2005	4500000	500000	'Jefe Mercadeo'	31.840.269	'3500'
1.751.219	'Melissa Roa'	'F'	19/06/60	16/03/2001	2250000	2500000	'Vendedor'	31.178.144	'2100'
768.782	'Joaquín Rosas'	'M'	07/07/47	16/05/1990	2250000	2500000	'Vendedor'	31.178.144	'2200'
737.689	'Mario Llano'	'M'	30/08/45	16/05/1990	2250000	2500000	'Vendedor'	31.178.144	'2300'
333.333.333	'Elisa Rojas'	'F'	28/09/79	01/06/2004	3000000	1000000	'Jefe Mecánicos'	31.840.269	'4000'
888.888	'Iván Duarte'	'M'	12/08/55	16/05/1998	1050000	200000	'Mecánico'	333.333.333	'4100'
11.111.111	'Irene Díaz'	'F'	28/09/79	01/06/2004	1050000	200000	'Mecánico'	333.333.333	'4200'
444.444	'Abel Gómez'	'M'	24/12/39	01/10/2000	1050000	200000	'Mecánico'	333.333.333	'4300'
1.130.222	'José Giraldo'	'M'	20/01/85	01/11/2000	1200000	400000	'Asesor'	22.222.222	'3500'
19.709.802	'William Daza'	'M'	09/10/82	16/12/1999	2250000	1000000	'Investigador'	16.759.060	'3000'
31.174.099	'Diana Solarte'	'F'	19/11/57	16/05/1990	1250000	500000	'Secretaria'	31.840.269	'1000'
1.130.777	'Marcos Cortez'	'M'	23/06/86	16/04/2000	2550000	500000	'Mecánico'	333.333.333	'4000'
1.130.782	'Antonio Gil'	'M'	23/01/80	16/04/2010	850000	1500000	'Técnico'	16.211.383	'1500'
333.333.334	'Marisol Pulido'	'F'	01/10/79	16/05/1990	3250000	1000000	'Investigador'	16.759.060	'3000'
333.333.335	'Ana Moreno'	'F'	05/01/92	01/06/2004	1200000	400000	'Secretaria'	16.759.060	'3000'
1.130.333	'Pedro Blanco'	'M'	28/10/87	01/10/2000	800000	3000000	'Vendedor'	31.178.144	'2000'
1.130.444	'Jesús Alfonso'	'M'	14/03/88	01/10/2000	800000	3500000	'Vendedor'	31.178.144	'2000'
333.333.336	'Carolina Ríos'	'F'	15/02/92	01/10/2000	1250000	500000	'Secretaria'	16.211.383	'1500'
333.333.337	'Edith Muñoz'	'F'	31/03/92	01/10/2000	800000	3600000	'Vendedor'	31.178.144	'2100'
1.130.555	'Julían Mora'	'M'	03/07/89	01/10/2000	800000	3100000	'Vendedor'	31.178.144	'2200'
1.130.666	'Manuel Millán'	'M'	08/12/90	01/06/2004	800000	3700000	'Vendedor'	31.178.144	'2300'



Con los modelos de datos de "Departamentos" y "Empleados" descritos antes se debe determinar:

- A. Los empleados del departamento 3000 van a recibir adicional a su pago normal un reconocimiento especial por única vez por valor de 500.000. Se necesita conocer cuál es el valor total que recibirán en su próximo pago, se espera poder visualizar la información ordenada alfabéticamente por empleado. El pago normal de un empleado incluye los valores de Salario y Comisión.

```
SELECT e.nomEmp, (e.salEmp + e.comisionE + 500000) AS 'Pago Total'  
FROM empleado e  
JOIN departamento d ON e.codDepto = d.CodDepto  
WHERE e.codDepto = 3000  
ORDER BY e.nomEmp;
```

- B. Se quiere conocer los departamentos que tienen más de 3 empleados y cuántos son esos empleados, se debe ordenar de mayor a menor por cantidad de empleados por departamento.
- C. Se quiere conocer si se tiene empleados que no estén asociados a alguno de los departamentos existentes.
- D. Se quiere conocer cuál es el departamento que tiene la nómina más costosa y cuál es el valor de la nómina. La nómina normal de un empleado incluye los valores de Salario y Comisión.

```
SELECT d.nombreDpto AS 'Departamento', SUM(e.salEmp + e.comisionE) AS 'Nómina Total'  
FROM empleado e  
JOIN departamento d ON e.codDepto = d.CodDepto  
GROUP BY d.nombreDpto  
ORDER BY SUM(e.salEmp + e.comisionE) DESC  
LIMIT 1;
```

- E. Se quiere conocer cuales son los 3 directores que tienen más personas a cargo.

```
SELECT jefeID, COUNT(*) AS 'Cantidad de Subordinados'  
FROM empleado  
GROUP BY jefeID  
ORDER BY COUNT(*) DESC  
LIMIT 3;
```



- F. Se quiere conocer el salario promedio de la empresa y el departamento con el salario promedio más alto y cual es ese salario. (obligatorio)

```
SELECT AVG(salEmp + comisionE) AS 'Salario Promedio Empresa'
```

```
FROM empleado;
```

```
SELECT d.nombreDpto AS 'Departamento', AVG(e.salEmp + e.comisionE) AS 'Salario Promedio'
```

```
FROM empleado e
```

```
JOIN departamento d ON e.codDpto = d.CodDpto
```

```
GROUP BY d.nombreDpto
```

```
ORDER BY AVG(e.salEmp + e.comisionE) DESC
```

```
LIMIT 1;
```

**IMPORTANTE:** Se deben responder al menos 4 consultas, la F es obligatoria.

2. Automatización Apis.

### **Criterios de aceptación:**

Se requiere crear un servicio para que los clientes puedan solicitar la recogida de sus productos, se solicitan que el campo **dirección** sea tipo alfanumérico y que sea obligatorio, el campo **fechaRecogida** debe ser de tipo alfanumérico en formato (yyyy-mm-dd) y obligatorio, se permite ingresar una fecha futura dentro de los 5 días hábiles siguientes, si se crea una solicitud con una fecha ya ingresada con la misma dirección no se permitirá realizar la solicitud de recogida, los campos **nombreEntrega**, **apellidosEntrega**, **celularEntrega**, **emailUsuario**, **descripcionTipoVia** y **aplicativo** son obligatorios y de tipo alfanumérico.

### **Información del servicio:**

**Tipo:** Post

#### **Url:**

<https://apiv2-test.coordinadora.com/recogidas/cm-solicitud-recogidas-ms/solicitud-recogida>

#### **Body:**

```
{  
  "tipoEnvio": "1",  
  "tipoProducto": "4",  
  "indicativo": "57",  
  "tipoDocumento": "13",  
  "email": "ana@gmail.com",
```

# COORDINADORA

```
"personaEntrega": "1",
"indicativoEntrega": "57",
"medidasAproximadas": [
  {
    "id": 2,
    "tipoPaq": "Par de zapatos",
    "nombrePaq": "Par de za...",
    "cantidad": 1
  }
],
"ciudad": "Envigado (Ant)",
"via": "",
"numero": "",
"detalle": "PARQUE SAN JOSE BOD 14",
"tipoVia": 16,
"nombres": "prueba",
"apellidos": "prueba1",
"documento": "1036149000",

"celular": "3005777777",
"ciudadDetalle": {
  "nombre_terminal_operativa": "Medellin",
  "tipo_servicio": "A",
  "dane_ciudad": "05266",
  "dane_actual": "05266000",
  "ciudad_tarifa": "05266000",
  "sms": true,
  "cubre_mqp": true,
  "codigo_postal": "055428",
  "terminal_operativa": 2,
  "cubre_me": true,
  "area_telefono": 4,
  "observaciones2": "FCE - RD - FD - RCE",
  "codigo": "05266000",
  "tipo_poblacion": "D",
  "activo": true,
  "codigo_terminal": 2,
  "codigo_interno": 204,
  "mensajeria": true,
  "cubre_mm": false,
  "departamento": "05",
  "cubre_cm": false,
  "nombre": "Envigado (Ant)",
  "abreviado": "ENVIGADO",
  "nombre_terminal": "Medellin",
```



```
"observaciones": ""  
},  
"direccion": "Cl 10 # 20 30",  
"fechaRecogida": "2023-12-04",  
"nombreEntrega": "prueba",  
"apellidosEntrega": "prueba1",  
"celularEntrega": "3045677777",  
"emailUsuario": "anar.7@gmail.com",  
"descripcionTipoVia": "Kilómetro",  
"aplicativo": "envios"  
}
```

Dado que se tiene el servicio con su información y todos los criterios de aceptación establecidos, se deben realizar las siguientes tareas:

- A). Se debe realizar una automatización con la herramienta de tu preferencia (Recomendado Playwright).
- B). Se debe diseñar mínimo 3 escenarios de prueba con escritura Gherkin con la herramienta de software Cucumber (BDD)
- C). Se debe validar del servicio, el mensaje y el código de respuesta según el escenario diseñado en la automatización.
- D). Se debe validar que los campos del servicio cumplan con las especificaciones y estándares definidos en los criterios de aceptación.
- E). Se debe generar un reporte en HTML de los resultados obtenidos de las pruebas.
- F). Se debe guardar la prueba en un repositorio de Github y compartir el link ([camilov@coordinadora.com](mailto:camilov@coordinadora.com), [manuel.patino@coordiandora.com](mailto:manuel.patino@coordiandora.com)).

#### Escenarios de prueba:

##### **Feature:** Solicitud de recogida

Como cliente

Quiero solicitar la recogida de mis productos

Para que sean recolectados por la empresa de mensajería

##### **Scenario:** Solicitud de recogida exitosa

**Given** Realizo una petición Post a la url

<https://apiv2-test.coordinadora.com/recogidas/cm-solicitud-recogidas-ms/solicitud-recogida>

**When** realizo una solicitud de recogida con una dirección válida y una fecha de recogida dentro de los próximos 5 días hábiles

**Then** la respuesta debe tener un código de estado 200



**And** la respuesta debe incluir un mensaje de éxito “Solicitud recogida programada exitosamente”

**And** la respuesta debe contener un código de seguimiento de la solicitud

The screenshot shows a REST client interface with a POST request to `https://apiv2-test.coordinadora.com/recogidas/cm-solicitud-recogidas-ms/solicitud-recogida`. The request body is a JSON object containing address and delivery details. The response is a 200 OK status with a JSON body indicating success.

```
POST https://apiv2-test.coordinadora.com/recogidas/cm-solicitud-recogidas-ms/solicitud-recogida

{
  "dane_ciudad": "11001",
  "nombre_terminal": "Bogota",
  "codigo_postal": "110831",
  "area_telefono": 1,
  "departamento": "25",
  "codigo_terminal": 1,
  "direccion": "Test376",
  "fechaRecogida": "2024-02-01",
  "nombreEntrega": "Nicolas",
  "apellidosEntrega": "Sanchez",
  "celularEntrega": "3204894961",
  "emailUsuario": "francisco.perez1@yopmail.com",
  "descripcionTipoVia": "Autopista",
  "aplicativo": "envios",
  "otraDireccion": "calle falsa 123",
  "aceptaTerminos": true
}
```

```
{
  "isError": false,
  "data": {
    "id_recogida": {
      "idRecogida": "26702078",
      "error": false,
      "message": "Solicitud recogida programada exitosamente"
    }
  },
  "timestamp": "2024-02-01T00:55:24.554Z",
  "id": "daca41724296eaa9e424fe3839e24325cd2a38b4"
}
```

**Feature:** Solicitud de recogida

Como cliente

Quiero solicitar la recogida de mis productos

Para que sean recolectados por la empresa de mensajería

**Scenario:** Solicitud de recogida con fecha de recogida pasada

**Given** que soy un cliente con productos para recoger

**When** intento realizar una solicitud de recogida con una fecha de recogida pasada

**Then** la respuesta debe tener un código de estado 200

**And** la respuesta debe incluir un mensaje de error indicando que “El campo fecha: fechaRecogida, no debe ser menor a la fecha actual”



The screenshot shows a REST client interface with a POST request to `https://api-v2-test.coordinadora.com/recogidas/cm-solicitud-recogidas-ms/solicitud-recogida`. The request body is a JSON object with the following fields: `direccion`, `fechaRecogida`, `nombreEntrega`, `apellidosEntrega`, `celularEntrega`, `emailUsuario`, `descripcionTipoVia`, `aplicativo`, `otraDireccion`, and `aceptaTerminos`. The response body is a JSON object with the following fields: `isError`, `data` (containing `message`, `idRecogidaAnterior`, and `recogida_anterior`), `timestamp`, and `id`.

```
POST https://api-v2-test.coordinadora.com/recogidas/cm-solicitud-recogidas-ms/solicitud-recogida

{
  "direccion": "test212",
  "fechaRecogida": "2023-02-01",
  "nombreEntrega": "Nicolas",
  "apellidosEntrega": "Sanchez",
  "celularEntrega": "3204894961",
  "emailUsuario": "francisco.perez1@yopmail.com",
  "descripcionTipoVia": "Autopista",
  "aplicativo": "envios",
  "otraDireccion": "calle falsa 123",
  "aceptaTerminos": true
}

{
  "isError": true,
  "data": {
    "message": "El campo fecha: 01-02-2023, no debe ser menor a la fecha actual.",
    "idRecogidaAnterior": "01-02-2023, no debe ser menor a la fecha actual.",
    "recogida_anterior": true
  },
  "timestamp": "2024-02-01T00:42:14.920Z",
  "id": "8599e147162b29e5b814d1c1bcc04e677c41b8df"
}
```

**Feature:** Validación de campos obligatorios en la solicitud de recogida

Como cliente

Quiero asegurarme de que los campos obligatorios en la solicitud de recogida estén correctamente validados

Para evitar errores y garantizar que la solicitud sea procesada correctamente

**Scenario:** Validación de campos obligatorios

**Given** que soy un cliente que desea solicitar la recogida de productos

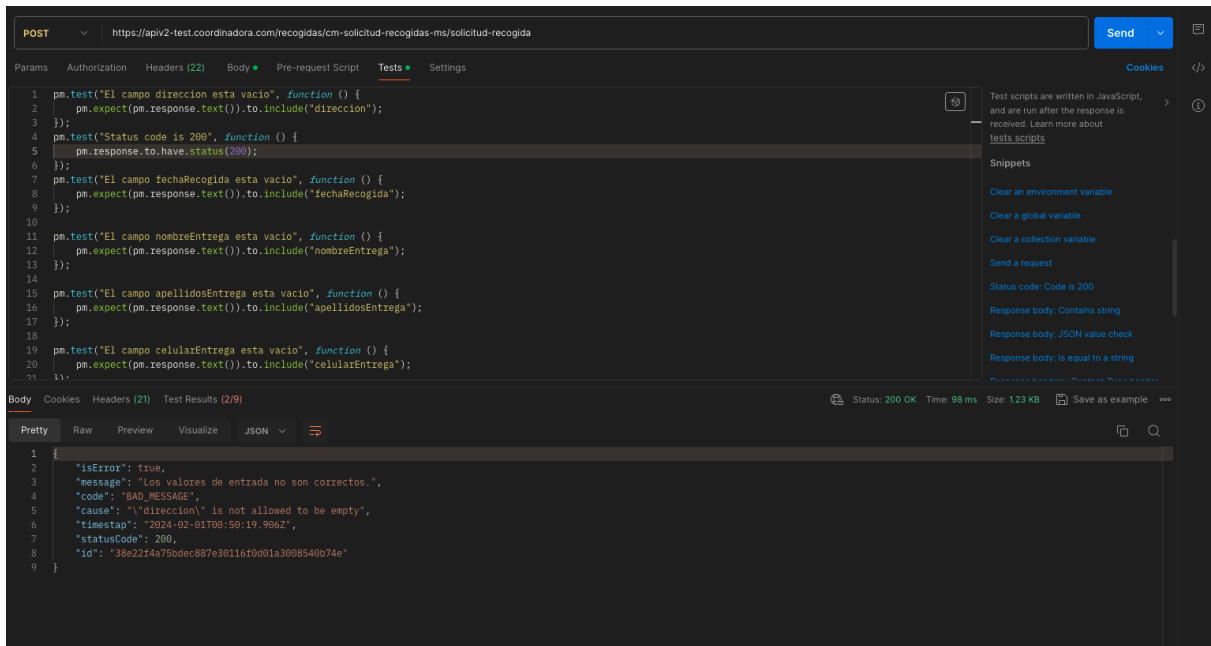
**When** intentó realizar una solicitud de recogida con campos obligatorios faltantes

**Then** la solicitud de recogida no se completa

**And** recibo mensajes de error indicando los campos faltantes



# COORDINADORA



**POST** `https://apiV2-test.coordinadora.com/recogidas/cm-solicitud-recogidas-ms/solicitud-recogida` **Send**

Params Authorization Headers (22) Body • Pre-request Script Tests • Settings Cookies </>

```
1 pm.test("El campo direccion esta vacio", function () {
2   pm.expect(pm.response.text()).to.include("direccion");
3 });
4 pm.test("Status code is 200", function () {
5   pm.response.to.have.status(200);
6 });
7 pm.test("El campo fechaRecogida esta vacio", function () {
8   pm.expect(pm.response.text()).to.include("fechaRecogida");
9 });
10
11 pm.test("El campo nombreEntrega esta vacio", function () {
12   pm.expect(pm.response.text()).to.include("nombreEntrega");
13 });
14
15 pm.test("El campo apellidosEntrega esta vacio", function () {
16   pm.expect(pm.response.text()).to.include("apellidosEntrega");
17 });
18
19 pm.test("El campo celularEntrega esta vacio", function () {
20   pm.expect(pm.response.text()).to.include("celularEntrega");
21 });
```

Test scripts are written in JavaScript, and are run after the response is received. Learn more about [tests scripts](#)

Snippets

- Clear an environment variable
- Clear a global variable
- Clear a collection variable
- Send a request
- Status code: Code is 200
- Response body: Contains string
- Response body: JSON value check
- Response body: Is equal to a string

Body Cookies Headers (21) Test Results (2/9) Status: 200 OK Time: 98 ms Size: 123 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "isError": true,
3   "message": "Los valores de entrada no son correctos.",
4   "code": "BAD_MESSAGE",
5   "cause": "\"direccion\" is not allowed to be empty",
6   "timestamp": "2024-02-01T08:59:19.996Z",
7   "statusCode": 200,
8   "id": "38e22f4a75bdec887e30116f0d01a3088540b74e"
9 }
```

## **Feature:** Solicitud de recogida

Como cliente

Quiero solicitar la recogida de mis productos

Para que sean recolectados por la empresa de mensajería

**Scenario:** Intento de solicitud de recogida para una dirección con recogida existente

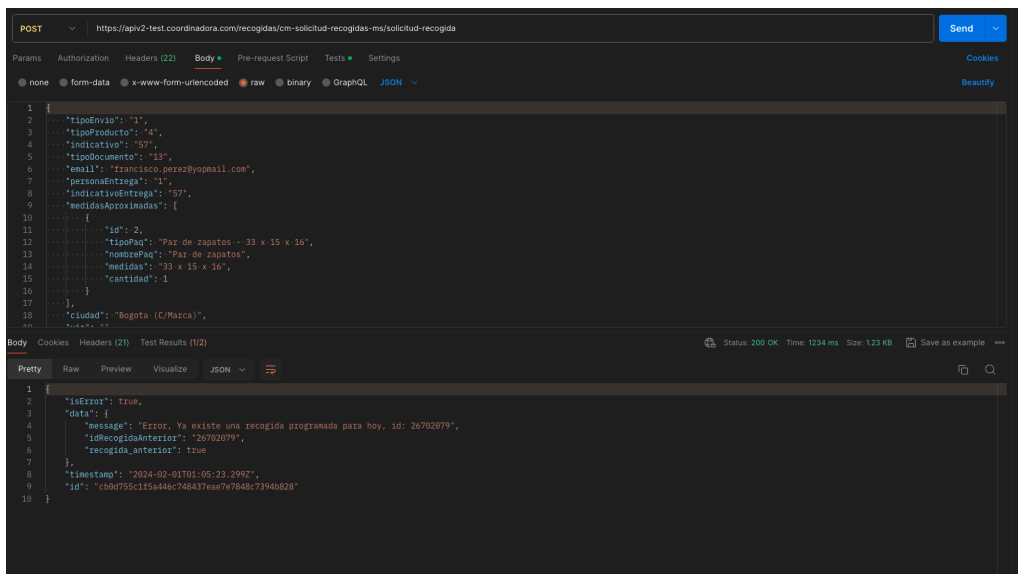
**Given** que soy un cliente con productos para recoger

**And** existe una recogida previamente programada para la misma dirección

**When** intento realizar una solicitud de recogida para la misma dirección

**Then** la solicitud de recogida no se completa

**And** recibo un mensaje de error indicando que ya existe una recogida programada para esa dirección



**POST** `https://apiV2-test.coordinadora.com/recogidas/cm-solicitud-recogidas-ms/solicitud-recogida` **Send**

Params Authorization Headers (22) Body • Pre-request Script Tests • Settings Cookies </>

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "tipoEnvio": "1",
3   "tipoProducto": "4",
4   "indicativo": "57",
5   "tipoDocumento": "13",
6   "email": "francisco.perez@yahoo.com",
7   "personaEntrega": "1",
8   "indicativoEntrega": "57",
9   "medidasAproximadas": [
10     {
11       "id": 2,
12       "tipoPaq": "Paq de zapatos - 33 x 15 x 16",
13       "nombrePaq": "Paq de zapatos",
14       "medidas": "33 x 15 x 16",
15       "cantidad": 1
16     }
17   ],
18   "ciudad": "Bogota (C/Marca)",
19 }
```

Body Cookies Headers (21) Test Results (1/2) Status: 200 OK Time: 1234 ms Size: 123 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "isError": true,
3   "data": {
4     "message": "Error, Ya existe una recogida programada para hoy, id: 26782679",
5     "idRecogidaAnterior": "26782679",
6     "recogida_anterior": true
7   },
8   "timestamp": "2024-02-01T01:05:23.299Z",
9   "id": "cb0d755c1f5a446c748437ee7e7848c7394b828"
10 }
```





### 3. Prueba de performance

**Url:**

<https://apiv2-test.coordinadora.com/recogidas/cm-solicitud-recogidas-ms/solicitud-recogida>

El proceso de solicitudes de recogidas tiene un comportamiento esperado por el negocio de manejar unas 9000 solicitudes al día, teniendo un 70% de estas entre las 7:30 am y 9:00 am, donde 1 usuario solo puede generar 1 solicitud de recogida a la vez. Para el proceso de solicitudes de recogidas se espera que tengan un crecimiento del 15% cada año. Se requiere realizar pruebas de performance que permitan validar si el servicio va poder responder a ese crecimiento proyectado para los próximos 5 años.

Para este caso yo sugiero realizar la prueba con la herramienta Jmeter ya que proporciona todo lo necesario para el desarrollo de esta, en esta herramienta podremos crear un test plan, y con el podremos agregar un grupo de hilos los cuales simulará la cantidad de usuario y la cantidad de interacciones que se realizarán en el API, también se pueden crear calendarios con el fin de programar la ejecución de la prueba en horas determinadas para poder cumplir con la prueba en horarios específicos.

# COORDINADORA

Coordinadora.jmx (/Users/nikolassanchez/Desktop/Coordinadora.jmx) - Apache JMeter (5.6.3)

00:11:47 0 0/9000

**Test Plan**

- Thread Group
  - Coordinadora
    - Summary Report
    - Aggregate Graph

**Thread Group**

Name: Thread Group

Comments:

Action to be taken after a Sampler error

☐ Continue ☒ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

**Thread Properties**

Number of Threads (users): 9000

Ramp-up period (seconds): 100

Loop Count: ☐ Infinite 5000

☒ Same user on each iteration

☐ Delay Thread creation until needed

☐ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):